

GDCM
2.2.0

Generated by Doxygen 1.8.1.1

Sun Jun 24 2012 03:55:18

Contents

1	GDCM Documentation	1
2	gdcm2pnm	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	12
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	Forcing (re)compression	19
5.6.11	Decompressing a Compressed DICOM	19
5.6.12	Compressing an uncompressed Icon	19
5.6.13	Generating an Icon	19
5.6.14	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	20
5.8.3	vbindiff	21
5.9	SEE ALSO	21
5.10	COPYRIGHT	21
6	gdcmdiff	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	26
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	35
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)	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	37
9.6.4	Convert RLE to DICOM	37
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	Warning	38
9.9	SEE ALSO	38
9.10	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	gdcmData	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 PDF to PDF/DICOM.	45
11.1 SYNOPSIS	45
11.2 DESCRIPTION	45
11.3 PARAMETERS	45
11.4 options	45
11.4.1 general options	45
11.5 Usage Example	46
11.6 PDF Info Mapping	46
11.7 SEE ALSO	47
11.8 COPYRIGHT	47
12 Extract Data Element Value Field.	49
12.1 SYNOPSIS	49
12.2 DESCRIPTION	49
12.3 PARAMETERS	49
12.4 options	49
12.4.1 PARAMETERS	49
12.4.2 options	49
12.4.3 general options	49
12.5 Typical usage	50
12.5.1 Copy Attribute Value to file	50
12.5.2 Extract Pixel Data	50
12.5.3 Encapsulated Syntax	50
12.5.4 Extract fragments as single file	51
12.6 Footnote about JPEG files	52
12.7 SEE ALSO	52
12.8 COPYRIGHT	52
13 Scan a directory containing DICOM files.	53
13.1 SYNOPSIS	53
13.2 DESCRIPTION	53
13.2.1 PARAMETERS	53
13.2.2 options	53
13.2.3 general options	53
13.3 Typical usage	54
13.4 Simple usage	54

13.5	Complex usage	54
13.6	SEE ALSO	54
13.7	COPYRIGHT	54
14	Tool to execute a DICOM Query/Retrieve operation	55
14.1	SYNOPSIS	55
14.2	DESCRIPTION	55
14.3	PARAMETERS	55
14.4	options	55
14.4.1	options	55
14.4.2	mode options	55
14.4.3	C-STORE options	56
14.4.4	C-FIND/C-MOVE options	56
14.4.5	C-MOVE options	56
14.4.6	general options	56
14.4.7	environment variable	56
14.5	C-ECHO usage	57
14.6	C-STORE usage	57
14.7	C-FIND usage	57
14.8	C-MOVE usage	58
14.9	patientroot notes	58
14.10	Debugging	58
14.11	Port Warning	58
14.12	C-STORE Warnings	59
14.13	C-MOVE Warnings	59
14.14	C-FIND IMAGE level (Composite Object Instance)	59
14.15	Storing the Query	59
14.16	SEE ALSO	60
14.17	COPYRIGHT	60
15	Concatenate/Extract DICOM files.	61
15.1	SYNOPSIS	61
15.2	DESCRIPTION	61
15.3	PARAMETERS	61
15.4	options	61
15.4.1	options	61
15.4.2	general options	61
15.4.3	environment variable	62

15.5 Typical usage	62
15.5.1 SIEMENS Mosaic	62
15.6 SEE ALSO	63
15.7 COPYRIGHT	63
16 Simple DICOM viewer.	65
16.1 SYNOPSIS	65
16.2 DESCRIPTION	65
16.3 PARAMETERS	65
16.4 options	65
16.4.1 options	65
16.4.2 general options	65
16.5 Typical usage	66
16.6 Simple usage	66
16.7 Wiki Link	66
16.8 SEE ALSO	66
16.9 COPYRIGHT	66
17 Todo List	67
18 Deprecated List	69
19 Bug List	71
20 Namespace Index	73
20.1 Namespace List	73
21 Class Index	75
21.1 Class Hierarchy	75
22 Class Index	83
22.1 Class List	83
23 File Index	97
23.1 File List	97
24 Namespace Documentation	103
24.1 gdcmm Namespace Reference	103
24.1.1 Detailed Description	116
24.1.2 Typedef Documentation	117
24.1.2.1 AECComp	117

24.1.2.2	ASComp	117
24.1.2.3	BOOL_FUNCTION_PFILE_PFILE_POINTER	117
24.1.2.4	CSComp	117
24.1.2.5	DAComp	117
24.1.2.6	DTComp	117
24.1.2.7	FileList	117
24.1.2.8	IconImage	117
24.1.2.9	LOComp	117
24.1.2.10	LTComp	117
24.1.2.11	MacroEntry	117
24.1.2.12	NestedMacroEntries	117
24.1.2.13	PNComp	117
24.1.2.14	SHComp	117
24.1.2.15	STComp	117
24.1.2.16	TMComp	117
24.1.2.17	UIComp	117
24.1.2.18	UTComp	117
24.1.3	Enumeration Type Documentation	117
24.1.3.1	CompOperators	117
24.1.3.2	ECharSet	118
24.1.3.3	EQueryLevel	118
24.1.3.4	EQueryType	119
24.1.3.5	ERootType	119
24.1.3.6	LodModeType	119
24.1.4	Function Documentation	119
24.1.4.1	backslash	119
24.1.4.2	GetVRFromTag	119
24.1.4.3	operator!=	119
24.1.4.4	operator!=	119
24.1.4.5	operator<<	119
24.1.4.6	operator<<	119
24.1.4.7	operator<<	119
24.1.4.8	operator<<	120
24.1.4.9	operator<<	120
24.1.4.10	operator<<	120
24.1.4.11	operator<<	120
24.1.4.12	operator<<	120

24.1.4.13 operator<<	120
24.1.4.14 operator<<	120
24.1.4.15 operator<<	120
24.1.4.16 operator<<	120
24.1.4.17 operator<<	120
24.1.4.18 operator<<	120
24.1.4.19 operator<<	120
24.1.4.20 operator<<	120
24.1.4.21 operator<<	120
24.1.4.22 operator<<	121
24.1.4.23 operator<<	121
24.1.4.24 operator<<	121
24.1.4.25 operator<<	121
24.1.4.26 operator<<	121
24.1.4.27 operator<<	121
24.1.4.28 operator<<	121
24.1.4.29 operator<<	121
24.1.4.30 operator<<	121
24.1.4.31 operator<<	121
24.1.4.32 operator<<	121
24.1.4.33 operator<<	121
24.1.4.34 operator<<	121
24.1.4.35 operator<<	121
24.1.4.36 operator<<	121
24.1.4.37 operator<<	121
24.1.4.38 operator<<	121
24.1.4.39 operator<<	122
24.1.4.40 operator<<	122
24.1.4.41 operator<<	122
24.1.4.42 operator<<	122
24.1.4.43 operator<<	122
24.1.4.44 operator<<	122
24.1.4.45 operator<<	122
24.1.4.46 operator<<	122
24.1.4.47 operator<<	122
24.1.4.48 operator<<	122
24.1.4.49 operator<<	122

24.1.4.50 operator<<	122
24.1.4.51 operator<<	123
24.1.4.52 operator<<	123
24.1.4.53 operator<<	123
24.1.4.54 operator<<	123
24.1.4.55 operator<<	123
24.1.4.56 operator<<	123
24.1.4.57 operator<<	123
24.1.4.58 operator==	123
24.1.4.59 operator>>	123
24.1.4.60 operator>>	123
24.1.4.61 operator>>	123
24.1.4.62 to_string	123
24.1.4.63 TYPETOENCODING	123
24.1.5 Variable Documentation	123
24.1.5.1 GlobalInstance	123
24.1.5.2 VRBINARY	124
24.2 gdcm::network Namespace Reference	124
24.2.1 Enumeration Type Documentation	127
24.2.1.1 EEventID	128
24.2.1.2 EStateID	128
24.2.2 Function Documentation	129
24.2.2.1 GetStateIndex	129
24.2.3 Variable Documentation	129
24.2.3.1 cMaxEventID	129
24.2.3.2 cMaxStateID	129
24.3 gdcm::SegmentHelper Namespace Reference	129
24.4 gdcm::terminal Namespace Reference	129
24.4.1 Detailed Description	130
24.4.2 Enumeration Type Documentation	130
24.4.2.1 Attribute	130
24.4.2.2 Color	130
24.4.2.3 Mode	130
24.4.3 Function Documentation	131
24.4.3.1 setattribute	131
24.4.3.2 setbgcolor	131
24.4.3.3 setfgcolor	131

24.4.3.4	setmode	131
24.5	itk Namespace Reference	131
25	Class Documentation	133
25.1	gdcm::network::AAabortPDU Class Reference	133
25.1.1	Detailed Description	134
25.1.2	Constructor & Destructor Documentation	134
25.1.2.1	AAabortPDU	134
25.1.3	Member Function Documentation	134
25.1.3.1	IsLastFragment	134
25.1.3.2	Print	134
25.1.3.3	Read	134
25.1.3.4	Size	135
25.1.3.5	Write	135
25.2	gdcm::network::AAssociateACPDU Class Reference	135
25.2.1	Detailed Description	136
25.2.2	Member Typedef Documentation	137
25.2.2.1	SizeType	137
25.2.3	Constructor & Destructor Documentation	137
25.2.3.1	AAssociateACPDU	137
25.2.4	Member Function Documentation	137
25.2.4.1	AddPresentationContextAC	137
25.2.4.2	GetNumberOfPresentationContextAC	137
25.2.4.3	GetPresentationContextAC	137
25.2.4.4	GetUserInfo	137
25.2.4.5	InitFromRQ	137
25.2.4.6	IsLastFragment	137
25.2.4.7	Print	137
25.2.4.8	Read	137
25.2.4.9	SetCalledAETitle	137
25.2.4.10	SetCallingAETitle	137
25.2.4.11	Size	137
25.2.4.12	Write	137
25.2.5	Friends And Related Function Documentation	137
25.2.5.1	AAssociateRQPDU	138
25.3	gdcm::network::AAssociateRJPDU Class Reference	138
25.3.1	Detailed Description	139

25.3.2	Constructor & Destructor Documentation	139
25.3.2.1	AAssociateRJPDU	139
25.3.3	Member Function Documentation	139
25.3.3.1	IsLastFragment	139
25.3.3.2	Print	139
25.3.3.3	Read	139
25.3.3.4	Size	139
25.3.3.5	Write	139
25.4	gdcm::network::AAssociateRQPDU Class Reference	139
25.4.1	Detailed Description	141
25.4.2	Member Typedef Documentation	141
25.4.2.1	PresentationContextArrayType	141
25.4.2.2	SizeType	141
25.4.3	Constructor & Destructor Documentation	141
25.4.3.1	AAssociateRQPDU	141
25.4.3.2	AAssociateRQPDU	141
25.4.4	Member Function Documentation	141
25.4.4.1	AddPresentationContext	141
25.4.4.2	GetCalledAETitle	141
25.4.4.3	GetCallingAETitle	141
25.4.4.4	GetNumberOfPresentationContext	141
25.4.4.5	GetPresentationContext	142
25.4.4.6	GetPresentationContextByAbstractSyntax	142
25.4.4.7	GetPresentationContextByID	142
25.4.4.8	GetPresentationContexts	142
25.4.4.9	IsAETitleValid	142
25.4.4.10	IsLastFragment	142
25.4.4.11	Print	142
25.4.4.12	Read	142
25.4.4.13	SetCalledAETitle	142
25.4.4.14	SetCallingAETitle	142
25.4.4.15	Size	142
25.4.4.16	Write	142
25.5	gdcm::AbortEvent Class Reference	143
25.6	gdcm::network::AbstractSyntax Class Reference	144
25.6.1	Detailed Description	144
25.6.2	Constructor & Destructor Documentation	144

25.6.2.1	AbstractSyntax	144
25.6.3	Member Function Documentation	144
25.6.3.1	GetAsDataElement	144
25.6.3.2	GetName	144
25.6.3.3	operator==	144
25.6.3.4	Print	144
25.6.3.5	Read	144
25.6.3.6	SetName	144
25.6.3.7	SetNameFromUID	144
25.6.3.8	Size	144
25.6.3.9	Write	144
25.7	gdcm::AnonymizeEvent Class Reference	145
25.7.1	Detailed Description	146
25.7.2	Member Typedef Documentation	146
25.7.2.1	Self	146
25.7.2.2	Superclass	146
25.7.3	Constructor & Destructor Documentation	146
25.7.3.1	AnonymizeEvent	146
25.7.3.2	~AnonymizeEvent	146
25.7.3.3	AnonymizeEvent	146
25.7.4	Member Function Documentation	146
25.7.4.1	CheckEvent	146
25.7.4.2	GetEventName	146
25.7.4.3	GetTag	146
25.7.4.4	MakeObject	146
25.7.4.5	SetTag	147
25.8	gdcm::Anonymizer Class Reference	147
25.8.1	Detailed Description	149
25.8.2	Constructor & Destructor Documentation	150
25.8.2.1	Anonymizer	150
25.8.2.2	~Anonymizer	150
25.8.3	Member Function Documentation	150
25.8.3.1	BALCPPProtect	150
25.8.3.2	BasicApplicationLevelConfidentialityProfile	150
25.8.3.3	CanEmptyTag	150
25.8.3.4	Empty	150
25.8.3.5	GetBasicApplicationLevelConfidentialityProfileAttributes	150

25.8.3.6	GetCryptographicMessageSyntax	150
25.8.3.7	GetFile	150
25.8.3.8	New	150
25.8.3.9	RecurseDataSet	151
25.8.3.10	Remove	151
25.8.3.11	RemoveGroupLength	151
25.8.3.12	RemovePrivateTags	151
25.8.3.13	RemoveRetired	151
25.8.3.14	Replace	151
25.8.3.15	Replace	151
25.8.3.16	SetCryptographicMessageSyntax	151
25.8.3.17	SetFile	152
25.9	gdcm::AnyEvent Class Reference	152
25.10	gdcm::network::ApplicationContext Class Reference	153
25.10.1	Detailed Description	153
25.10.2	Constructor & Destructor Documentation	153
25.10.2.1	ApplicationContext	153
25.10.3	Member Function Documentation	154
25.10.3.1	GetName	154
25.10.3.2	Print	154
25.10.3.3	Read	154
25.10.3.4	SetName	154
25.10.3.5	Size	154
25.10.3.6	Write	154
25.11	gdcm::ApplicationEntity Class Reference	154
25.11.1	Detailed Description	155
25.11.2	Member Function Documentation	155
25.11.2.1	IsValid	155
25.11.2.2	Print	155
25.11.2.3	SetBlob	155
25.11.2.4	Squeeze	155
25.11.3	Member Data Documentation	155
25.11.3.1	Internal	155
25.11.3.2	MaxLength	155
25.11.3.3	MaxNumberOfComponents	155
25.11.3.4	Padding	155
25.11.3.5	Separator	156

25.12gdcn::network::AReleaseRPPDU Class Reference	156
25.12.1 Detailed Description	157
25.12.2 Constructor & Destructor Documentation	157
25.12.2.1 AReleaseRPPDU	157
25.12.3 Member Function Documentation	157
25.12.3.1 IsLastFragment	157
25.12.3.2 Print	157
25.12.3.3 Read	157
25.12.3.4 Size	157
25.12.3.5 Write	157
25.13gdcn::network::AReleaseRQPDU Class Reference	157
25.13.1 Detailed Description	158
25.13.2 Constructor & Destructor Documentation	158
25.13.2.1 AReleaseRQPDU	159
25.13.3 Member Function Documentation	159
25.13.3.1 IsLastFragment	159
25.13.3.2 Print	159
25.13.3.3 Read	159
25.13.3.4 Size	159
25.13.3.5 Write	159
25.14gdcn::network::ARTIMTimer Class Reference	159
25.14.1 Detailed Description	160
25.14.2 Constructor & Destructor Documentation	160
25.14.2.1 ARTIMTimer	160
25.14.3 Member Function Documentation	160
25.14.3.1 GetElapsedTime	160
25.14.3.2 GetHasExpired	160
25.14.3.3 GetTimeout	160
25.14.3.4 SetTimeout	160
25.14.3.5 Start	160
25.14.3.6 Stop	160
25.15gdcn::ASN1 Class Reference	160
25.15.1 Detailed Description	161
25.15.2 Constructor & Destructor Documentation	161
25.15.2.1 ASN1	161
25.15.2.2 ~ASN1	161
25.15.3 Member Function Documentation	161

25.15.3.1 ParseDump	161
25.15.3.2 ParseDumpFile	161
25.15.3.3 TestPBKDF2	161
25.16gdcmm::network::AsynchronousOperationsWindowSub Class Reference	161
25.16.1 Detailed Description	161
25.16.2 Constructor & Destructor Documentation	162
25.16.2.1 AsynchronousOperationsWindowSub	162
25.16.3 Member Function Documentation	162
25.16.3.1 Read	162
25.16.3.2 Size	162
25.16.3.3 Write	162
25.17gdcmm::Attribute< Group, Element, TVR, TVM > Class Template Reference	162
25.17.1 Detailed Description	163
25.17.2 Member Typedef Documentation	164
25.17.2.1 ArrayType	164
25.17.3 Member Enumeration Documentation	164
25.17.3.1 anonymous enum	164
25.17.4 Member Function Documentation	164
25.17.4.1 GDCM_STATIC_ASSERT	164
25.17.4.2 GDCM_STATIC_ASSERT	164
25.17.4.3 GDCM_STATIC_ASSERT	164
25.17.4.4 GetAsDataElement	164
25.17.4.5 GetDictVM	164
25.17.4.6 GetDictVR	165
25.17.4.7 GetNumberOfValues	165
25.17.4.8 GetTag	165
25.17.4.9 GetValue	165
25.17.4.10GetValue	165
25.17.4.11GetValues	166
25.17.4.12GetVM	166
25.17.4.13GetVR	166
25.17.4.14operator!=	166
25.17.4.15operator<	166
25.17.4.16operator==	166
25.17.4.17operator[]	167
25.17.4.18operator[]	167
25.17.4.19Print	167

25.17.4.20Set	167
25.17.4.21SetByteValue	167
25.17.4.22SetByteValueNoSwap	167
25.17.4.23SetFromDataElement	168
25.17.4.24SetFromDataSet	168
25.17.4.25SetValue	168
25.17.4.26SetValues	168
25.17.5 Member Data Documentation	168
25.17.5.1 Internal	168
25.18gdcmm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference	169
25.18.1 Member Typedef Documentation	170
25.18.1.1 ArrayType	170
25.18.2 Member Enumeration Documentation	170
25.18.2.1 anonymous enum	170
25.18.3 Member Function Documentation	171
25.18.3.1 GDCM_STATIC_ASSERT	171
25.18.3.2 GDCM_STATIC_ASSERT	171
25.18.3.3 GDCM_STATIC_ASSERT	171
25.18.3.4 GDCM_STATIC_ASSERT	171
25.18.3.5 GetAsDataElement	171
25.18.3.6 GetDictVM	171
25.18.3.7 GetDictVR	171
25.18.3.8 GetNumberOfValues	171
25.18.3.9 GetTag	171
25.18.3.10GetValue	171
25.18.3.11GetValue	171
25.18.3.12GetValues	171
25.18.3.13GetVM	172
25.18.3.14GetVR	172
25.18.3.15operator!=	172
25.18.3.16operator<	172
25.18.3.17operator==	172
25.18.3.18Print	172
25.18.3.19Set	172
25.18.3.20SetByteValue	172
25.18.3.21SetByteValueNoSwap	172
25.18.3.22SetFromDataElement	173

25.18.3.23SetFromDataSet	173
25.18.3.24SetValue	173
25.18.4 Member Data Documentation	173
25.18.4.1 Internal	173
25.19gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference	173
25.19.1 Member Function Documentation	174
25.19.1.1 GetVM	174
25.20gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference	175
25.20.1 Member Function Documentation	176
25.20.1.1 GetVM	176
25.21gdcmm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference	176
25.21.1 Member Typedef Documentation	177
25.21.1.1 ArrayType	177
25.21.2 Constructor & Destructor Documentation	177
25.21.2.1 Attribute	177
25.21.2.2 ~Attribute	177
25.21.3 Member Function Documentation	177
25.21.3.1 GDCM_STATIC_ASSERT	177
25.21.3.2 GDCM_STATIC_ASSERT	177
25.21.3.3 GDCM_STATIC_ASSERT	178
25.21.3.4 GetAsDataElement	178
25.21.3.5 GetDictVM	178
25.21.3.6 GetDictVR	178
25.21.3.7 GetNumberOfValues	178
25.21.3.8 GetTag	178
25.21.3.9 GetValue	178
25.21.3.10GetValue	178
25.21.3.11GetValues	178
25.21.3.12GetVM	178
25.21.3.13GetVR	179
25.21.3.14operator[]	179
25.21.3.15operator[]	179
25.21.3.16Print	179
25.21.3.17SetByteValue	179
25.21.3.18SetFromDataElement	179
25.21.3.19SetNumberOfValues	179
25.21.3.20SetValue	179

25.21.3.21SetValue	179
25.21.3.22SetValues	180
25.22gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	180
25.22.1 Member Function Documentation	181
25.22.1.1 GetVM	181
25.23gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	181
25.23.1 Member Function Documentation	183
25.23.1.1 GetVM	183
25.24gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference	183
25.24.1 Member Function Documentation	184
25.24.1.1 GetVM	184
25.25gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference	184
25.25.1 Member Function Documentation	186
25.25.1.1 GetVM	186
25.26gdcmm::AudioCodec Class Reference	186
25.26.1 Detailed Description	187
25.26.2 Constructor & Destructor Documentation	187
25.26.2.1 AudioCodec	187
25.26.2.2 ~AudioCodec	187
25.26.3 Member Function Documentation	187
25.26.3.1 CanCode	187
25.26.3.2 CanDecode	188
25.26.3.3 Decode	188
25.27gdcmm::Base64 Class Reference	188
25.27.1 Detailed Description	188
25.27.2 Constructor & Destructor Documentation	188
25.27.2.1 Base64	188
25.27.2.2 ~Base64	188
25.27.3 Member Function Documentation	188
25.27.3.1 Decode	189
25.27.3.2 Encode	189
25.27.3.3 GetDecodeLength	189
25.27.3.4 GetEncodeLength	189
25.28gdcmm::network::BaseCompositeMessage Class Reference	189
25.28.1 Detailed Description	190
25.28.2 Member Function Documentation	191
25.28.2.1 ConstructPDV	191

25.29gdcm::network::BasePDU Class Reference	191
25.29.1 Detailed Description	192
25.29.2 Constructor & Destructor Documentation	192
25.29.2.1 ~BasePDU	192
25.29.3 Member Function Documentation	192
25.29.3.1 IsLastFragment	192
25.29.3.2 Print	192
25.29.3.3 Read	192
25.29.3.4 Size	193
25.29.3.5 Write	193
25.30gdcm::BaseRootQuery Class Reference	193
25.30.1 Constructor & Destructor Documentation	195
25.30.1.1 BaseRootQuery	195
25.30.1.2 ~BaseRootQuery	195
25.30.2 Member Function Documentation	195
25.30.2.1 AddQueryDataSet	195
25.30.2.2 GetAbstractSyntaxUID	195
25.30.2.3 GetQueryDataSet	195
25.30.2.4 GetQueryDataSet	195
25.30.2.5 GetTagListByLevel	195
25.30.2.6 InitializeDataSet	195
25.30.2.7 SetSearchParameter	195
25.30.2.8 SetSearchParameter	195
25.30.2.9 SetSearchParameter	195
25.30.2.10ValidateQuery	196
25.30.2.11WriteHelpFile	196
25.30.2.12WriteQuery	196
25.30.3 Friends And Related Function Documentation	196
25.30.3.1 QueryFactory	196
25.30.4 Member Data Documentation	196
25.30.4.1 mDataSet	196
25.30.4.2 mHelpDescription	196
25.30.4.3 mImage	196
25.30.4.4 mPatient	196
25.30.4.5 mRootType	196
25.30.4.6 mSeries	196
25.30.4.7 mStudy	196

25.31gdcmm::SegmentHelper::BasicCodedEntry Struct Reference	196
25.31.1 Detailed Description	198
25.31.2 Constructor & Destructor Documentation	198
25.31.2.1 BasicCodedEntry	198
25.31.2.2 BasicCodedEntry	198
25.31.2.3 BasicCodedEntry	198
25.31.3 Member Function Documentation	198
25.31.3.1 IsEmpty	198
25.31.4 Member Data Documentation	198
25.31.4.1 CM	198
25.31.4.2 CSD	198
25.31.4.3 CSV	198
25.31.4.4 CV	199
25.32gdcmm::BasicOffsetTable Class Reference	199
25.32.1 Detailed Description	200
25.32.2 Constructor & Destructor Documentation	200
25.32.2.1 BasicOffsetTable	200
25.32.3 Member Function Documentation	200
25.32.3.1 Read	201
25.32.4 Friends And Related Function Documentation	201
25.32.4.1 operator<<	201
25.33gdcmm::Bitmap Class Reference	201
25.33.1 Detailed Description	204
25.33.2 Member Typedef Documentation	204
25.33.2.1 LUTPtr	204
25.33.3 Constructor & Destructor Documentation	204
25.33.3.1 Bitmap	204
25.33.3.2 ~Bitmap	204
25.33.4 Member Function Documentation	204
25.33.4.1 AreOverlaysInPixelData	204
25.33.4.2 Clear	204
25.33.4.3 ComputeLossyFlag	204
25.33.4.4 GetBuffer	204
25.33.4.5 GetBuffer2	204
25.33.4.6 GetBufferLength	204
25.33.4.7 GetColumns	205
25.33.4.8 GetDataElement	205

25.33.4.9 GetDataElement	205
25.33.4.10GetDimension	205
25.33.4.11GetDimensions	205
25.33.4.12GetLUT	205
25.33.4.13GetLUT	205
25.33.4.14GetNeedByteSwap	205
25.33.4.15GetNumberOfDimensions	205
25.33.4.16GetPhotometricInterpretation	205
25.33.4.17GetPixelFormat	206
25.33.4.18GetPixelFormat	206
25.33.4.19GetPlanarConfiguration	206
25.33.4.20GetRows	206
25.33.4.21GetTransferSyntax	206
25.33.4.22IsEmpty	206
25.33.4.23IsLossy	206
25.33.4.24IsTransferSyntaxCompatible	206
25.33.4.25Print	206
25.33.4.26SetColumns	206
25.33.4.27SetDataElement	206
25.33.4.28SetDimension	207
25.33.4.29SetDimensions	207
25.33.4.30SetLossyFlag	207
25.33.4.31SetLUT	207
25.33.4.32SetNeedByteSwap	207
25.33.4.33SetNumberOfDimensions	207
25.33.4.34SetPhotometricInterpretation	207
25.33.4.35SetPixelFormat	207
25.33.4.36SetPlanarConfiguration	207
25.33.4.37SetRows	208
25.33.4.38SetTransferSyntax	208
25.33.4.39TryJPEG2000Codec	208
25.33.4.40TryJPEG2000Codec2	208
25.33.4.41TryJPEGCodec	208
25.33.4.42TryJPEGCodec2	208
25.33.4.43TryJPEGLSCodec	208
25.33.4.44TryKAKADUCodec	208
25.33.4.45TryPVRGCodec	208

25.33.4.46TryRAWCodec	208
25.33.4.47TryRLECodec	208
25.33.5 Friends And Related Function Documentation	208
25.33.5.1 ImageChangeTransferSyntax	208
25.33.5.2 PixmapReader	208
25.33.6 Member Data Documentation	208
25.33.6.1 Dimensions	208
25.33.6.2 LossyFlag	208
25.33.6.3 LUT	208
25.33.6.4 NeedByteSwap	208
25.33.6.5 NumberOfDimensions	208
25.33.6.6 PF	208
25.33.6.7 PI	208
25.33.6.8 PixelData	208
25.33.6.9 PlanarConfiguration	209
25.33.6.10TS	209
25.34gdcm::BitmapToBitmapFilter Class Reference	209
25.34.1 Detailed Description	210
25.34.2 Constructor & Destructor Documentation	210
25.34.2.1 BitmapToBitmapFilter	210
25.34.2.2 ~BitmapToBitmapFilter	211
25.34.3 Member Function Documentation	211
25.34.3.1 GetOutput	211
25.34.3.2 SetInput	211
25.34.4 Member Data Documentation	211
25.34.4.1 Input	211
25.34.4.2 Output	211
25.35gdcm::ByteBuffer Class Reference	211
25.35.1 Detailed Description	211
25.35.2 Constructor & Destructor Documentation	212
25.35.2.1 ByteBuffer	212
25.35.3 Member Function Documentation	212
25.35.3.1 Get	212
25.35.3.2 GetStart	212
25.35.3.3 ShiftEnd	212
25.35.3.4 UpdatePosition	212
25.36gdcm::ByteSwap< T > Class Template Reference	212

25.36.1 Detailed Description	212
25.36.2 Member Function Documentation	212
25.36.2.1 Swap	212
25.36.2.2 SwapFromSwapCodeIntoSystem	213
25.36.2.3 SwapRange	213
25.36.2.4 SwapRangeFromSwapCodeIntoSystem	213
25.36.2.5 SystemIsBigEndian	213
25.36.2.6 SystemIsLittleEndian	213
25.37gdcm::ByteSwapFilter Class Reference	213
25.37.1 Detailed Description	213
25.37.2 Constructor & Destructor Documentation	214
25.37.2.1 ByteSwapFilter	214
25.37.2.2 ~ByteSwapFilter	214
25.37.3 Member Function Documentation	214
25.37.3.1 ByteSwap	214
25.37.3.2 SetByteSwapTag	214
25.38gdcm::ByteValue Class Reference	214
25.38.1 Detailed Description	216
25.38.2 Constructor & Destructor Documentation	216
25.38.2.1 ByteValue	216
25.38.2.2 ByteValue	216
25.38.2.3 ~ByteValue	216
25.38.3 Member Function Documentation	216
25.38.3.1 Clear	216
25.38.3.2 Fill	216
25.38.3.3 GetBuffer	216
25.38.3.4 GetLength	217
25.38.3.5 GetPointer	217
25.38.3.6 IsEmpty	217
25.38.3.7 IsPrintable	217
25.38.3.8 operator const std::vector< char > &	217
25.38.3.9 operator=	217
25.38.3.10operator==	217
25.38.3.11operator==	217
25.38.3.12Print	217
25.38.3.13PrintASCII	218
25.38.3.14PrintGroupLength	218

25.38.3.15PrintHex	218
25.38.3.16Read	218
25.38.3.17Read	218
25.38.3.18SetLength	218
25.38.3.19Write	218
25.38.3.20Write	218
25.38.3.21WriteBuffer	218
25.39gdcmm::network::CEchoRQ Class Reference	218
25.39.1 Detailed Description	219
25.39.2 Member Function Documentation	220
25.39.2.1 ConstructPDV	220
25.39.3 Member Data Documentation	220
25.39.3.1 AffectedSOPClassUID	220
25.39.3.2 MessageID	220
25.40gdcmm::network::CEchoRSP Class Reference	220
25.40.1 Member Function Documentation	221
25.40.1.1 ConstructPDV	221
25.41gdcmm::network::CFind Class Reference	221
25.41.1 Detailed Description	221
25.42gdcmm::network::CFindCancelRQ Class Reference	221
25.42.1 Member Function Documentation	222
25.42.1.1 ConstructPDV	222
25.43gdcmm::network::CFindRQ Class Reference	223
25.43.1 Member Function Documentation	223
25.43.1.1 ConstructPDV	224
25.44gdcmm::network::CFindRSP Class Reference	224
25.44.1 Member Function Documentation	225
25.44.1.1 ConstructPDV	225
25.45gdcmm::network::CMoveCancelRq Class Reference	225
25.45.1 Member Function Documentation	226
25.45.1.1 ConstructPDV	226
25.46gdcmm::network::CMoveRQ Class Reference	226
25.46.1 Detailed Description	227
25.46.2 Member Function Documentation	227
25.46.2.1 ConstructPDV	228
25.47gdcmm::network::CMoveRSP Class Reference	228
25.47.1 Detailed Description	229

25.47.2 Member Function Documentation	229
25.47.2.1 ConstructPDV	229
25.48gdcmm::Codec Class Reference	229
25.48.1 Detailed Description	230
25.49gdcmm::Coder Class Reference	230
25.49.1 Detailed Description	231
25.49.2 Constructor & Destructor Documentation	231
25.49.2.1 ~Coder	231
25.49.3 Member Function Documentation	231
25.49.3.1 CanCode	231
25.49.3.2 Code	231
25.49.3.3 InternalCode	231
25.50gdcmm::CodeString Class Reference	232
25.50.1 Detailed Description	233
25.50.2 Member Typedef Documentation	233
25.50.2.1 const_iterator	233
25.50.2.2 const_reference	233
25.50.2.3 const_reverse_iterator	233
25.50.2.4 difference_type	233
25.50.2.5 iterator	233
25.50.2.6 pointer	233
25.50.2.7 reference	233
25.50.2.8 reverse_iterator	233
25.50.2.9 size_type	233
25.50.2.10value_type	233
25.50.3 Constructor & Destructor Documentation	233
25.50.3.1 CodeString	233
25.50.3.2 CodeString	233
25.50.3.3 CodeString	233
25.50.3.4 CodeString	233
25.50.4 Member Function Documentation	234
25.50.4.1 GetAsString	234
25.50.4.2 IsValid	234
25.50.4.3 Size	234
25.50.4.4 TrimInternal	234
25.50.5 Friends And Related Function Documentation	234
25.50.5.1 operator!=	234

25.50.5.2 operator<<	234
25.50.5.3 operator==	234
25.51gdcmm::Command Class Reference	234
25.51.1 Detailed Description	236
25.51.2 Constructor & Destructor Documentation	236
25.51.2.1 Command	236
25.51.2.2 ~Command	236
25.51.3 Member Function Documentation	236
25.51.3.1 Execute	236
25.51.3.2 Execute	236
25.52gdcmm::CommandDataSet Class Reference	236
25.52.1 Detailed Description	238
25.52.2 Constructor & Destructor Documentation	238
25.52.2.1 CommandDataSet	238
25.52.2.2 ~CommandDataSet	238
25.52.3 Member Function Documentation	238
25.52.3.1 Insert	238
25.52.3.2 Read	238
25.52.3.3 Replace	238
25.52.3.4 Write	238
25.52.4 Friends And Related Function Documentation	238
25.52.4.1 operator<<	239
25.53gdcmm::network::CompositeMessageFactory Class Reference	239
25.53.1 Detailed Description	239
25.53.2 Member Function Documentation	239
25.53.2.1 ConstructCEchoRQ	239
25.53.2.2 ConstructCFindRQ	239
25.53.2.3 ConstructCMoveRQ	239
25.53.2.4 ConstructCStoreRQ	240
25.53.2.5 ConstructCStoreRSP	240
25.54gdcmm::CompositeNetworkFunctions Class Reference	240
25.54.1 Detailed Description	240
25.54.2 Member Typedef Documentation	241
25.54.2.1 KeyValuePairArrayType	241
25.54.2.2 KeyValuePairType	241
25.54.3 Member Function Documentation	241
25.54.3.1 CEcho	241

25.54.3.2 CFind	241
25.54.3.3 CMove	242
25.54.3.4 ConstructQuery	242
25.54.3.5 ConstructQuery	242
25.54.3.6 CStore	242
25.55gdcM::ConstCharWrapper Class Reference	243
25.55.1 Detailed Description	243
25.55.2 Constructor & Destructor Documentation	243
25.55.2.1 ConstCharWrapper	243
25.55.3 Member Function Documentation	243
25.55.3.1 operator const char *	243
25.56gdcM::CP246ExplicitDataElement Class Reference	243
25.56.1 Detailed Description	245
25.56.2 Member Function Documentation	245
25.56.2.1 GetLength	245
25.56.2.2 Read	245
25.56.2.3 ReadPreValue	245
25.56.2.4 ReadValue	245
25.56.2.5 ReadWithLength	245
25.57gdcM::CryptographicMessageSyntax Class Reference	245
25.57.1 Detailed Description	246
25.57.2 Member Enumeration Documentation	246
25.57.2.1 CipherTypes	246
25.57.3 Constructor & Destructor Documentation	246
25.57.3.1 CryptographicMessageSyntax	246
25.57.3.2 ~CryptographicMessageSyntax	246
25.57.4 Member Function Documentation	246
25.57.4.1 Decrypt	246
25.57.4.2 Encrypt	246
25.57.4.3 GetCipherType	247
25.57.4.4 ParseCertificateFile	247
25.57.4.5 ParseKeyFile	247
25.57.4.6 SetCipherType	247
25.58gdcM::CSAElement Class Reference	247
25.58.1 Detailed Description	248
25.58.2 Member Typedef Documentation	249
25.58.2.1 DataPtr	249

25.58.3 Constructor & Destructor Documentation	249
25.58.3.1 CSAElement	249
25.58.3.2 CSAElement	249
25.58.4 Member Function Documentation	249
25.58.4.1 GetByteValue	249
25.58.4.2 GetKey	249
25.58.4.3 GetName	249
25.58.4.4 GetNoOfItems	249
25.58.4.5 GetSyngoDT	249
25.58.4.6 GetValue	249
25.58.4.7 GetValue	250
25.58.4.8 GetVM	250
25.58.4.9 GetVR	250
25.58.4.10 IsEmpty	250
25.58.4.11 operator<	250
25.58.4.12 operator=	250
25.58.4.13 operator==	250
25.58.4.14 SetByteValue	250
25.58.4.15 SetKey	250
25.58.4.16 SetName	250
25.58.4.17 SetNoOfItems	250
25.58.4.18 SetSyngoDT	250
25.58.4.19 SetValue	250
25.58.4.20 SetVM	250
25.58.4.21 SetVR	250
25.58.5 Friends And Related Function Documentation	251
25.58.5.1 operator<<	251
25.58.6 Member Data Documentation	251
25.58.6.1 DataField	251
25.58.6.2 KeyField	251
25.58.6.3 NameField	251
25.58.6.4 NoOfItemsField	251
25.58.6.5 SyngoDTField	251
25.58.6.6 ValueMultiplicityField	251
25.58.6.7 VRField	251
25.59gdcm::CSAHeader Class Reference	251
25.59.1 Detailed Description	253

25.59.2 Member Enumeration Documentation	253
25.59.2.1 CSAHeaderType	253
25.59.3 Constructor & Destructor Documentation	253
25.59.3.1 CSAHeader	253
25.59.3.2 ~CSAHeader	253
25.59.4 Member Function Documentation	253
25.59.4.1 FindCSAElementByName	254
25.59.4.2 GetCSADataInfo	254
25.59.4.3 GetCSAEEnd	254
25.59.4.4 GetCSAElementByName	254
25.59.4.5 GetCSAImageHeaderInfoTag	254
25.59.4.6 GetCSASeriesHeaderInfoTag	254
25.59.4.7 GetDataSet	255
25.59.4.8 GetFormat	255
25.59.4.9 GetInterfile	255
25.59.4.10 LoadFromDataElement	255
25.59.4.11 Print	255
25.59.4.12 Read	255
25.59.4.13 Write	255
25.59.5 Friends And Related Function Documentation	255
25.59.5.1 operator<<	255
25.60 gdcmm::CSAHeaderDict Class Reference	255
25.60.1 Detailed Description	256
25.60.2 Member Typedef Documentation	256
25.60.2.1 ConstIterator	256
25.60.2.2 Iterator	256
25.60.2.3 MapCSAHeaderDictEntry	256
25.60.3 Constructor & Destructor Documentation	256
25.60.3.1 CSAHeaderDict	256
25.60.4 Member Function Documentation	256
25.60.4.1 AddCSAHeaderDictEntry	257
25.60.4.2 Begin	257
25.60.4.3 End	257
25.60.4.4 GetCSAHeaderDictEntry	257
25.60.4.5 IsEmpty	257
25.60.4.6 LoadDefault	257
25.60.5 Friends And Related Function Documentation	257

25.60.5.1 Dicts	257
25.60.5.2 operator<<	257
25.61gdcM::CSAHeaderDictEntry Class Reference	257
25.61.1 Detailed Description	258
25.61.2 Constructor & Destructor Documentation	258
25.61.2.1 CSAHeaderDictEntry	258
25.61.3 Member Function Documentation	258
25.61.3.1 GetDescription	258
25.61.3.2 GetName	258
25.61.3.3 GetVM	258
25.61.3.4 GetVR	258
25.61.3.5 operator<	259
25.61.3.6 SetDescription	259
25.61.3.7 SetName	259
25.61.3.8 SetVM	259
25.61.3.9 SetVR	259
25.61.4 Friends And Related Function Documentation	259
25.61.4.1 operator<<	259
25.62gdcM::CSAHeaderDictException Class Reference	259
25.63gdcM::network::CStoreRQ Class Reference	260
25.63.1 Detailed Description	261
25.63.2 Member Function Documentation	261
25.63.2.1 ConstructPDV	261
25.64gdcM::network::CStoreRSP Class Reference	261
25.64.1 Member Function Documentation	262
25.64.1.1 ConstructPDV	262
25.65gdcM::Curve Class Reference	263
25.65.1 Detailed Description	264
25.65.2 Constructor & Destructor Documentation	264
25.65.2.1 Curve	264
25.65.2.2 ~Curve	264
25.65.2.3 Curve	264
25.65.3 Member Function Documentation	264
25.65.3.1 Decode	264
25.65.3.2 GetAsPoints	264
25.65.3.3 GetDataValueRepresentation	265
25.65.3.4 GetDimensions	265

25.65.3.5 GetGroup	265
25.65.3.6 GetNumberOfCurves	265
25.65.3.7 GetNumberOfPoints	265
25.65.3.8 GetTypeInfoData	265
25.65.3.9 GetTypeInfoDataDescription	265
25.65.3.10IsEmpty	265
25.65.3.11Print	265
25.65.3.12SetCoordinateStartValue	265
25.65.3.13SetCoordinateStepValue	265
25.65.3.14SetCurve	265
25.65.3.15SetCurveDataDescriptor	265
25.65.3.16SetCurveDescription	265
25.65.3.17SetDataValueRepresentation	265
25.65.3.18SetDimensions	265
25.65.3.19SetGroup	265
25.65.3.20SetNumberOfPoints	265
25.65.3.21SetTypeInfoData	265
25.65.3.22Update	265
25.66gdcm::DataElement Class Reference	265
25.66.1 Detailed Description	268
25.66.2 Member Typedef Documentation	268
25.66.2.1 ValuePtr	268
25.66.3 Constructor & Destructor Documentation	268
25.66.3.1 DataElement	268
25.66.3.2 DataElement	269
25.66.4 Member Function Documentation	269
25.66.4.1 Clear	269
25.66.4.2 Empty	269
25.66.4.3 GetByteValue	269
25.66.4.4 GetLength	269
25.66.4.5 GetSequenceOfFragments	269
25.66.4.6 GetSequenceOfItems	270
25.66.4.7 GetSequenceOfItems	270
25.66.4.8 GetTag	270
25.66.4.9 GetTag	270
25.66.4.10GetValue	270
25.66.4.11GetValue	270

25.66.4.12	GetValueAsSQ	270
25.66.4.13	GetVL	271
25.66.4.14	GetVL	271
25.66.4.15	GetVR	271
25.66.4.16	IsEmpty	271
25.66.4.17	IsUndefinedLength	271
25.66.4.18	operator<	272
25.66.4.19	operator=	272
25.66.4.20	operator==	272
25.66.4.21	Read	272
25.66.4.22	ReadOrSkip	272
25.66.4.23	ReadPreValue	272
25.66.4.24	ReadValue	272
25.66.4.25	ReadWithLength	272
25.66.4.26	SetByteValue	272
25.66.4.27	SetTag	273
25.66.4.28	SetValue	273
25.66.4.29	SetVL	273
25.66.4.30	SetVLToUndefined	273
25.66.4.31	SetVR	273
25.66.4.32	Write	274
25.66.5	Friends And Related Function Documentation	274
25.66.5.1	operator<<	274
25.66.6	Member Data Documentation	274
25.66.6.1	TagField	274
25.66.6.2	ValueField	274
25.66.6.3	ValueLengthField	274
25.66.6.4	VRField	274
25.67	gdcm::DataElementException Class Reference	275
25.68	gdcm::DataEvent Class Reference	275
25.68.1	Detailed Description	277
25.68.2	Member Typedef Documentation	277
25.68.2.1	Self	277
25.68.2.2	Superclass	277
25.68.3	Constructor & Destructor Documentation	277
25.68.3.1	DataEvent	277
25.68.3.2	~DataEvent	277

25.68.3.3 DataEvent	277
25.68.4 Member Function Documentation	277
25.68.4.1 CheckEvent	277
25.68.4.2 GetData	277
25.68.4.3 GetDataLength	277
25.68.4.4 GetEventName	277
25.68.4.5 MakeObject	277
25.68.4.6 SetData	277
25.69gdcm::DataSet Class Reference	278
25.69.1 Detailed Description	280
25.69.2 Member Typedef Documentation	280
25.69.2.1 ConstIterator	280
25.69.2.2 DataElementSet	280
25.69.2.3 Iterator	280
25.69.2.4 SizeType	280
25.69.3 Member Function Documentation	280
25.69.3.1 Begin	280
25.69.3.2 Begin	280
25.69.3.3 Clear	280
25.69.3.4 ComputeDataElement	281
25.69.3.5 ComputeGroupLength	281
25.69.3.6 End	281
25.69.3.7 End	281
25.69.3.8 FindDataElement	281
25.69.3.9 FindDataElement	281
25.69.3.10FindNextDataElement	281
25.69.3.11GetDataElement	281
25.69.3.12GetDataElement	282
25.69.3.13GetDEEnd	282
25.69.3.14GetDES	282
25.69.3.15GetDES	282
25.69.3.16GetLength	282
25.69.3.17GetPrivateCreator	282
25.69.3.18Insert	282
25.69.3.19InsertDataElement	282
25.69.3.20IsEmpty	283
25.69.3.21operator()	283

25.69.3.22operator=	283
25.69.3.23operator[]	283
25.69.3.24Print	283
25.69.3.25Read	283
25.69.3.26ReadNested	283
25.69.3.27ReadSelectedTags	283
25.69.3.28ReadSelectedTagsWithLength	283
25.69.3.29ReadUpToTag	283
25.69.3.30ReadUpToTagWithLength	283
25.69.3.31ReadWithLength	283
25.69.3.32Remove	283
25.69.3.33Replace	283
25.69.3.34ReplaceEmpty	284
25.69.3.35Size	284
25.69.3.36Write	284
25.69.4 Friends And Related Function Documentation	284
25.69.4.1 CSAHeader	284
25.69.4.2 operator<<	284
25.70gdcm::DataSetEvent Class Reference	284
25.70.1 Detailed Description	286
25.70.2 Member Typedef Documentation	286
25.70.2.1 Self	286
25.70.2.2 Superclass	286
25.70.3 Constructor & Destructor Documentation	286
25.70.3.1 DataSetEvent	286
25.70.3.2 ~DataSetEvent	286
25.70.3.3 DataSetEvent	286
25.70.4 Member Function Documentation	286
25.70.4.1 CheckEvent	286
25.70.4.2 GetDataSet	286
25.70.4.3 GetEventName	286
25.70.4.4 MakeObject	286
25.71gdcm::DataSetHelper Class Reference	287
25.71.1 Detailed Description	287
25.71.2 Member Function Documentation	287
25.71.2.1 ComputeVR	287
25.72gdcm::Decoder Class Reference	287

25.72.1 Detailed Description	288
25.72.2 Constructor & Destructor Documentation	288
25.72.2.1 ~Decoder	288
25.72.3 Member Function Documentation	288
25.72.3.1 CanDecode	288
25.72.3.2 Decode	288
25.72.3.3 Decode	288
25.73gdcm::DefinedTerms Class Reference	289
25.73.1 Detailed Description	289
25.73.2 Constructor & Destructor Documentation	289
25.73.2.1 DefinedTerms	289
25.74gdcm::Defs Class Reference	289
25.74.1 Detailed Description	290
25.74.2 Constructor & Destructor Documentation	290
25.74.2.1 Defs	290
25.74.2.2 ~Defs	290
25.74.3 Member Function Documentation	290
25.74.3.1 GetIODFromFile	290
25.74.3.2 GetIODNameFromMediaStorage	290
25.74.3.3 GetIODs	291
25.74.3.4 GetIODs	291
25.74.3.5 GetMacros	291
25.74.3.6 GetMacros	291
25.74.3.7 GetModules	291
25.74.3.8 GetModules	291
25.74.3.9 GetTypeFromTag	291
25.74.3.10IsEmpty	291
25.74.3.11LoadDefaults	291
25.74.3.12LoadFromFile	291
25.74.3.13Verify	291
25.74.3.14Verify	291
25.74.4 Friends And Related Function Documentation	291
25.74.4.1 Global	291
25.75gdcm::DeltaEncodingCodec Class Reference	291
25.75.1 Detailed Description	293
25.75.2 Constructor & Destructor Documentation	293
25.75.2.1 DeltaEncodingCodec	293

25.75.2.2 ~DeltaEncodingCodec	293
25.75.3 Member Function Documentation	293
25.75.3.1 CanDecode	293
25.75.3.2 Decode	293
25.75.3.3 Decode	293
25.76gdcmm::DICOMDIR Class Reference	293
25.76.1 Detailed Description	293
25.76.2 Constructor & Destructor Documentation	294
25.76.2.1 DICOMDIR	294
25.76.2.2 DICOMDIR	294
25.77gdcmm::DICOMDIRGenerator Class Reference	294
25.77.1 Detailed Description	295
25.77.2 Member Typedef Documentation	295
25.77.2.1 FilenamesType	295
25.77.2.2 FilenameType	295
25.77.3 Constructor & Destructor Documentation	295
25.77.3.1 DICOMDIRGenerator	295
25.77.3.2 ~DICOMDIRGenerator	295
25.77.4 Member Function Documentation	295
25.77.4.1 AddImageDirectoryRecord	295
25.77.4.2 AddPatientDirectoryRecord	295
25.77.4.3 AddSeriesDirectoryRecord	295
25.77.4.4 AddStudyDirectoryRecord	295
25.77.4.5 Generate	295
25.77.4.6 GetFile	296
25.77.4.7 GetScanner	296
25.77.4.8 SetDescriptor	296
25.77.4.9 SetFile	296
25.77.4.10SetFilenames	296
25.77.4.11SetRootDirectory	296
25.78gdcmm::Dict Class Reference	296
25.78.1 Detailed Description	297
25.78.2 Member Typedef Documentation	297
25.78.2.1 ConstIterator	297
25.78.2.2 Iterator	297
25.78.2.3 MapDictEntry	297
25.78.3 Constructor & Destructor Documentation	297

25.78.3.1 Dict	297
25.78.4 Member Function Documentation	297
25.78.4.1 AddDictEntry	297
25.78.4.2 Begin	297
25.78.4.3 End	298
25.78.4.4 GetDictEntry	298
25.78.4.5 GetDictEntryByKeyword	298
25.78.4.6 GetDictEntryByName	298
25.78.4.7 GetKeywordFromTag	298
25.78.4.8 IsEmpty	298
25.78.4.9 LoadDefault	298
25.78.5 Friends And Related Function Documentation	298
25.78.5.1 Dicts	298
25.78.5.2 operator<<	298
25.79gdcmm::DictConverter Class Reference	299
25.79.1 Detailed Description	299
25.79.2 Member Enumeration Documentation	300
25.79.2.1 OutputTypes	300
25.79.3 Constructor & Destructor Documentation	300
25.79.3.1 DictConverter	300
25.79.3.2 ~DictConverter	300
25.79.4 Member Function Documentation	300
25.79.4.1 AddGroupLength	300
25.79.4.2 Convert	300
25.79.4.3 ConvertToCXX	300
25.79.4.4 ConvertToXML	300
25.79.4.5 GetDictName	300
25.79.4.6 GetInputFilename	300
25.79.4.7 GetOutputFilename	300
25.79.4.8 GetOutputType	300
25.79.4.9 Readuint16	300
25.79.4.10ReadVM	300
25.79.4.11ReadVR	300
25.79.4.12SetDictName	300
25.79.4.13SetInputFileName	300
25.79.4.14SetOutputFileName	300
25.79.4.15SetOutputType	300

25.79.4.16WriteFooter	301
25.79.4.17WriteHeader	301
25.80gdcm::DictEntry Class Reference	301
25.80.1 Detailed Description	302
25.80.2 Constructor & Destructor Documentation	302
25.80.2.1 DictEntry	302
25.80.3 Member Function Documentation	302
25.80.3.1 GetKeyword	302
25.80.3.2 GetName	302
25.80.3.3 GetRetired	302
25.80.3.4 GetVM	302
25.80.3.5 GetVR	303
25.80.3.6 IsUnique	303
25.80.3.7 SetElementXX	303
25.80.3.8 SetGroupXX	303
25.80.3.9 SetKeyword	303
25.80.3.10SetName	303
25.80.3.11SetRetired	303
25.80.3.12SetVM	303
25.80.3.13SetVR	303
25.80.4 Friends And Related Function Documentation	303
25.80.4.1 operator<<	303
25.81gdcm::DictPrinter Class Reference	303
25.81.1 Detailed Description	305
25.81.2 Constructor & Destructor Documentation	305
25.81.2.1 DictPrinter	305
25.81.2.2 ~DictPrinter	305
25.81.3 Member Function Documentation	305
25.81.3.1 Print	305
25.81.3.2 PrintDataElement2	305
25.81.3.3 PrintDataSet2	305
25.82gdcm::Dicts Class Reference	305
25.82.1 Detailed Description	306
25.82.2 Member Enumeration Documentation	306
25.82.2.1 ConstructorType	306
25.82.3 Constructor & Destructor Documentation	307
25.82.3.1 Dicts	307

25.82.3.2 ~Dicts	307
25.82.4 Member Function Documentation	307
25.82.4.1 GetConstructorString	307
25.82.4.2 GetCSAHeaderDict	307
25.82.4.3 GetDictEntry	307
25.82.4.4 GetDictEntry	307
25.82.4.5 GetPrivateDict	307
25.82.4.6 GetPrivateDict	307
25.82.4.7 GetPublicDict	307
25.82.4.8 IsEmpty	307
25.82.4.9 LoadDefaults	307
25.82.5 Friends And Related Function Documentation	307
25.82.5.1 Global	307
25.82.5.2 operator<<	308
25.83gdcm::network::DIMSE Class Reference	308
25.83.1 Detailed Description	308
25.83.2 Member Enumeration Documentation	308
25.83.2.1 CommandTypes	309
25.84gdcm::DirectionCosines Class Reference	309
25.84.1 Detailed Description	310
25.84.2 Constructor & Destructor Documentation	310
25.84.2.1 DirectionCosines	310
25.84.2.2 DirectionCosines	310
25.84.2.3 ~DirectionCosines	310
25.84.3 Member Function Documentation	310
25.84.3.1 ComputeDistAlongNormal	310
25.84.3.2 Cross	310
25.84.3.3 CrossDot	310
25.84.3.4 Dot	311
25.84.3.5 IsValid	311
25.84.3.6 Normalize	311
25.84.3.7 operator const double *	311
25.84.3.8 Print	311
25.84.3.9 SetFromString	311
25.85gdcm::Directory Class Reference	311
25.85.1 Detailed Description	312
25.85.2 Member Typedef Documentation	312

25.85.2.1	FileNamesType	312
25.85.2.2	FilenameType	313
25.85.3	Constructor & Destructor Documentation	313
25.85.3.1	Directory	313
25.85.3.2	~Directory	313
25.85.4	Member Function Documentation	313
25.85.4.1	Explore	313
25.85.4.2	GetDirectories	313
25.85.4.3	GetFileNames	313
25.85.4.4	GetToplevel	313
25.85.4.5	Load	313
25.85.4.6	Print	313
25.85.5	Friends And Related Function Documentation	314
25.85.5.1	operator<<	314
25.86	gdcmm::DirectoryHelper Class Reference	314
25.86.1	Member Function Documentation	314
25.86.1.1	GetCTImageSeriesUIDs	314
25.86.1.2	GetFileNamesFromSeriesUIDs	314
25.86.1.3	GetFrameOfReference	315
25.86.1.4	GetMRImageSeriesUIDs	315
25.86.1.5	GetRTStructSeriesUIDs	315
25.86.1.6	GetSeriesUIDsBySOPClassUID	315
25.86.1.7	GetSOPClassUID	315
25.86.1.8	LoadImageFromFiles	315
25.86.1.9	RetrieveSOPInstanceUIDFromIndex	315
25.86.1.10	RetrieveSOPInstanceUIDFromZPosition	315
25.87	gdcmm::DummyValueGenerator Class Reference	315
25.87.1	Detailed Description	315
25.87.2	Member Function Documentation	316
25.87.2.1	Generate	316
25.88	gdcmm::Dumper Class Reference	316
25.88.1	Detailed Description	317
25.88.2	Constructor & Destructor Documentation	317
25.88.2.1	Dumper	317
25.88.2.2	~Dumper	317
25.89	gdcmm::Element< TVR, TVM > Class Template Reference	318
25.89.1	Detailed Description	319

25.89.2 Member Typedef Documentation	319
25.89.2.1 Type	319
25.89.3 Member Function Documentation	319
25.89.3.1 GetAsDataElement	319
25.89.3.2 GetLength	319
25.89.3.3 GetValue	319
25.89.3.4 GetValue	320
25.89.3.5 GetValues	320
25.89.3.6 GetVM	320
25.89.3.7 GetVR	320
25.89.3.8 operator[]	320
25.89.3.9 Print	320
25.89.3.10Read	320
25.89.3.11Set	320
25.89.3.12SetFromDataElement	320
25.89.3.13SetNoSwap	320
25.89.3.14SetValue	320
25.89.3.15Write	320
25.89.4 Member Data Documentation	320
25.89.4.1 Internal	321
25.90gdcm::Element< TVR, VM::VM1_2 > Class Template Reference	321
25.90.1 Member Typedef Documentation	322
25.90.1.1 Parent	322
25.90.2 Member Function Documentation	322
25.90.2.1 SetLength	322
25.91gdcm::Element< TVR, VM::VM1_n > Class Template Reference	322
25.91.1 Member Typedef Documentation	324
25.91.1.1 Type	324
25.91.2 Constructor & Destructor Documentation	324
25.91.2.1 Element	324
25.91.2.2 ~Element	324
25.91.2.3 Element	324
25.91.3 Member Function Documentation	324
25.91.3.1 GetAsDataElement	324
25.91.3.2 GetLength	324
25.91.3.3 GetValue	324
25.91.3.4 GetValue	324

25.91.3.5 GetVM	324
25.91.3.6 GetVR	324
25.91.3.7 operator=	324
25.91.3.8 operator[]	324
25.91.3.9 Print	324
25.91.3.10Read	325
25.91.3.11Set	325
25.91.3.12SetArray	325
25.91.3.13SetFromDataElement	325
25.91.3.14SetLength	325
25.91.3.15SetNoSwap	325
25.91.3.16SetValue	325
25.91.3.17Write	325
25.91.3.18WriteASCII	325
25.92gdcmm::Element< TVR, VM::VM2_2n > Class Template Reference	325
25.92.1 Member Typedef Documentation	327
25.92.1.1 Parent	327
25.92.2 Member Function Documentation	327
25.92.2.1 SetLength	327
25.93gdcmm::Element< TVR, VM::VM2_n > Class Template Reference	327
25.93.1 Member Typedef Documentation	329
25.93.1.1 Parent	329
25.93.2 Member Function Documentation	329
25.93.2.1 SetLength	329
25.94gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference	329
25.94.1 Member Typedef Documentation	330
25.94.1.1 Parent	330
25.94.2 Member Function Documentation	330
25.94.2.1 SetLength	330
25.95gdcmm::Element< TVR, VM::VM3_n > Class Template Reference	331
25.95.1 Member Typedef Documentation	332
25.95.1.1 Parent	332
25.95.2 Member Function Documentation	332
25.95.2.1 SetLength	332
25.96gdcmm::Element< VR::AS, VM::VM5 > Class Template Reference	332
25.96.1 Member Function Documentation	332
25.96.1.1 GetLength	332

25.96.1.2 Print	332
25.96.2 Member Data Documentation	332
25.96.2.1 Internal	333
25.97gdcmm::Element< VR::OB, VM::VM1 > Class Template Reference	333
25.98gdcmm::Element< VR::OW, VM::VM1 > Class Template Reference	334
25.99gdcmm::EncapsulatedDocument Class Reference	336
25.99.1 Detailed Description	336
25.99.2 Constructor & Destructor Documentation	336
25.99.2.1 EncapsulatedDocument	336
25.100gdcmm::EncodingImplementation< VR::VRASCII > Class Template Reference	336
25.100.1 Member Function Documentation	337
25.100.1.1 Read	337
25.100.1.2 ReadComputeLength	337
25.100.1.3 ReadNoSwap	337
25.100.1.4 Write	337
25.100.1.5 Write	337
25.100.1.6 Write	337
25.101gdcmm::EncodingImplementation< VR::VRBINARY > Class Template Reference	337
25.101.1 Member Function Documentation	338
25.101.1.1 Read	338
25.101.1.2 ReadComputeLength	338
25.101.1.3 ReadNoSwap	338
25.101.1.4 Write	338
25.102gdcmm::EndEvent Class Reference	338
25.103gdcmm::EnumeratedValues Class Reference	339
25.103.1 Detailed Description	339
25.103.2 Constructor & Destructor Documentation	340
25.103.2.1 EnumeratedValues	340
25.104gdcmm::Event Class Reference	340
25.104.1 Detailed Description	341
25.104.2 Constructor & Destructor Documentation	341
25.104.2.1 Event	341
25.104.2.2 Event	341
25.104.2.3 Event	341
25.104.3 Member Function Documentation	341
25.104.3.1 CheckEvent	341
25.104.3.2 GetEventName	341

25.104.3.3	MakeObject	341
25.104.3.4	Print	341
25.105	dcm::Exception Class Reference	342
25.105.1	Detailed Description	343
25.105.2	Constructor & Destructor Documentation	343
25.105.2.1	Exception	343
25.105.2.2	~Exception	343
25.105.3	Member Function Documentation	343
25.105.3.1	GetDescription	343
25.105.3.2	what	343
25.106	dcm::ExitEvent Class Reference	343
25.107	dcm::ExplicitDataElement Class Reference	344
25.107.1	Detailed Description	346
25.107.2	Member Function Documentation	346
25.107.2.1	GetLength	346
25.107.2.2	Read	346
25.107.2.3	ReadPreValue	346
25.107.2.4	ReadValue	346
25.107.2.5	ReadWithLength	346
25.107.2.6	Write	346
25.108	dcm::ExplicitImplicitDataElement Class Reference	346
25.108.1	Detailed Description	348
25.108.2	Member Function Documentation	348
25.108.2.1	GetLength	348
25.108.2.2	Read	348
25.108.2.3	ReadPreValue	348
25.108.2.4	ReadValue	348
25.108.2.5	ReadWithLength	348
25.109	dcm::Fiducials Class Reference	348
25.109.1	Detailed Description	349
25.109.2	Constructor & Destructor Documentation	349
25.109.2.1	Fiducials	349
25.110	dcm::File Class Reference	349
25.110.1	Detailed Description	350
25.110.2	Constructor & Destructor Documentation	351
25.110.2.1	File	351
25.110.2.2	~File	351

25.110.3	Member Function Documentation	351
25.110.3.1	GetDataSet	351
25.110.3.2	GetDataSet	351
25.110.3.3	GetHeader	351
25.110.3.4	GetHeader	352
25.110.3.5	Read	352
25.110.3.6	SetDataSet	352
25.110.3.7	SetHeader	352
25.110.3.8	Write	352
25.110.4	Friends And Related Function Documentation	352
25.110.4.1	operator<<	352
25.111	dcm::FileDerivation Class Reference	352
25.111.1	Detailed Description	353
25.111.2	Constructor & Destructor Documentation	353
25.111.2.1	FileDerivation	353
25.111.2.2	~FileDerivation	353
25.111.3	Member Function Documentation	353
25.111.3.1	AddDerivationDescription	353
25.111.3.2	AddPurposeOfReferenceCodeSequence	353
25.111.3.3	AddReference	353
25.111.3.4	AddSourceImageSequence	354
25.111.3.5	Derive	354
25.111.3.6	GetFile	354
25.111.3.7	GetFile	354
25.111.3.8	SetDerivationCodeSequenceCodeValue	354
25.111.3.9	SetDerivationDescription	354
25.111.3.10	SetFile	354
25.111.3.11	SetPurposeOfReferenceCodeSequenceCodeValue	354
25.112	dcm::FileExplicitFilter Class Reference	355
25.112.1	Detailed Description	355
25.112.2	Constructor & Destructor Documentation	356
25.112.2.1	FileExplicitFilter	356
25.112.2.2	~FileExplicitFilter	356
25.112.3	Member Function Documentation	356
25.112.3.1	Change	356
25.112.3.2	ChangeFMI	356
25.112.3.3	GetFile	356

25.112.3.4	ProcessDataSet	356
25.112.3.5	SetChangePrivateTags	356
25.112.3.6	SetFile	356
25.112.3.7	SetRecomputeItemLength	356
25.112.3.8	SetRecomputeSequenceLength	356
25.112.3.9	SetUseVRUN	356
25.113	gdcm::FileMetaInformation Class Reference	357
25.113.1	Detailed Description	359
25.113.2	Constructor & Destructor Documentation	359
25.113.2.1	FileMetaInformation	359
25.113.2.2	~FileMetaInformation	359
25.113.2.3	FileMetaInformation	359
25.113.3	Member Function Documentation	359
25.113.3.1	AppendImplementationClassUID	359
25.113.3.2	ComputeDataSetMediaStorageSOPClass	359
25.113.3.3	ComputeDataSetTransferSyntax	359
25.113.3.4	Default	359
25.113.3.5	FillFromDataSet	359
25.113.3.6	GetDataSetTransferSyntax	360
25.113.3.7	GetFileMetaInformationVersion	360
25.113.3.8	GetFullLength	360
25.113.3.9	GetGDCMImplementationClassUID	360
25.113.3.10	GetGDCMImplementationVersionName	360
25.113.3.11	GetGDCMSourceApplicationEntityTitle	360
25.113.3.12	GetImplementationClassUID	360
25.113.3.13	GetImplementationVersionName	360
25.113.3.14	GetMediaStorage	360
25.113.3.15	GetMetaInformationTS	360
25.113.3.16	GetPreamble	360
25.113.3.17	GetPreamble	360
25.113.3.18	GetSourceApplicationEntityTitle	360
25.113.3.19	Insert	360
25.113.3.20	Valid	361
25.113.3.21	Read	361
25.113.3.22	ReadCompat	361
25.113.3.23	ReadCompatInternal	361
25.113.3.24	Replace	361

25.113.3.25	DataSetTransferSyntax	361
25.113.3.26	DataSetImplementationClassUID	361
25.113.3.27	DataSetImplementationVersionName	361
25.113.3.28	DataSetPreamble	361
25.113.3.29	DataSetSourceApplicationEntityTitle	361
25.113.3.30	DataSetWrite	361
25.113.4	Friends And Related Function Documentation	362
25.113.4.1	operator<<	362
25.113.5	Member Data Documentation	362
25.113.5.1	DataSetMS	362
25.113.5.2	DataSetTS	362
25.113.5.3	MetalInformationTS	362
25.114	dcm::Filename Class Reference	362
25.114.1	Detailed Description	363
25.114.2	Constructor & Destructor Documentation	363
25.114.2.1	Filename	363
25.114.3	Member Function Documentation	363
25.114.3.1	GetExtension	363
25.114.3.2	GetFileName	363
25.114.3.3	GetName	363
25.114.3.4	GetPath	363
25.114.3.5	IsEmpty	363
25.114.3.6	IsIdentical	363
25.114.3.7	Join	363
25.114.3.8	operator const char *	363
25.114.3.9	ToUnixSlashes	364
25.114.3.10	ToWindowsSlashes	364
25.115	dcm::FilenameGenerator Class Reference	364
25.115.1	Detailed Description	365
25.115.2	Member Typedef Documentation	365
25.115.2.1	FileNamesType	365
25.115.2.2	FilenameType	365
25.115.2.3	SizeType	365
25.115.3	Constructor & Destructor Documentation	365
25.115.3.1	FilenameGenerator	365
25.115.3.2	~FilenameGenerator	365
25.115.4	Member Function Documentation	365

25.115.4.1Generate	365
25.115.4.2GetFilename	365
25.115.4.3GetFileNames	365
25.115.4.4GetNumberOfFileNames	366
25.115.4.5GetPattern	366
25.115.4.6GetPrefix	366
25.115.4.7SetNumberOfFileNames	366
25.115.4.8SetPattern	366
25.115.4.9SetPrefix	366
25.116dcm::FileSet Class Reference	366
25.116.1Detailed Description	367
25.116.2Member Typedef Documentation	367
25.116.2.1FileType	367
25.116.2.2FileType	367
25.116.3Constructor & Destructor Documentation	367
25.116.3.1FileSet	367
25.116.4Member Function Documentation	367
25.116.4.1AddFile	367
25.116.4.2AddFile	367
25.116.4.3GetFiles	367
25.116.4.4SetFiles	367
25.116.5Friends And Related Function Documentation	367
25.116.5.1operator<<	367
25.117dcm::FileWithName Class Reference	368
25.117.1Detailed Description	369
25.117.2Constructor & Destructor Documentation	369
25.117.2.1FileWithName	369
25.117.3Member Data Documentation	369
25.117.3.1filename	369
25.118dcm::FindPatientRootQuery Class Reference	369
25.118.1Detailed Description	371
25.118.2Constructor & Destructor Documentation	371
25.118.2.1FindPatientRootQuery	371
25.118.3Member Function Documentation	371
25.118.3.1GetAbstractSyntaxUID	371
25.118.3.2GetTagListByLevel	371
25.118.3.3InitializeDataSet	371

25.118.3.4	ValidateQuery	371
25.118.4	Friends And Related Function Documentation	371
25.118.4.1	QueryFactory	371
25.119	gdcm::FindStudyRootQuery Class Reference	372
25.119.1	Detailed Description	373
25.119.2	Constructor & Destructor Documentation	373
25.119.2.1	FindStudyRootQuery	373
25.119.3	Member Function Documentation	373
25.119.3.1	GetAbstractSyntaxUID	373
25.119.3.2	GetTagListByLevel	373
25.119.3.3	InitializeDataSet	373
25.119.3.4	ValidateQuery	373
25.119.4	Friends And Related Function Documentation	373
25.119.4.1	QueryFactory	374
25.120	gdcm::Fragment Class Reference	374
25.120.1	Detailed Description	375
25.120.2	Constructor & Destructor Documentation	376
25.120.2.1	Fragment	376
25.120.3	Member Function Documentation	376
25.120.3.1	GetLength	376
25.120.3.2	Read	376
25.120.3.3	ReadValue	376
25.120.3.4	Write	376
25.120.4	Friends And Related Function Documentation	376
25.120.4.1	operator<<	376
25.121	itk::GDCMImageIO2 Class Reference	376
25.121.1	Detailed Description	379
25.121.2	Member Typedef Documentation	379
25.121.2.1	Pointer	379
25.121.2.2	Self	379
25.121.2.3	Superclass	379
25.121.3	Member Enumeration Documentation	379
25.121.3.1	ITCompressionType	379
25.121.4	Constructor & Destructor Documentation	380
25.121.4.1	GDCMImageIO2	380
25.121.4.2	~GDCMImageIO2	380
25.121.5	Member Function Documentation	380

25.121.5.1CanReadFile	380
25.121.5.2CanWriteFile	380
25.121.5.3GetBodyPart	380
25.121.5.4GetInstitution	380
25.121.5.5GetLabelFromTag	380
25.121.5.6GetLoadPrivateTagsDefault	380
25.121.5.7GetLoadSequencesDefault	380
25.121.5.8GetManufacturer	380
25.121.5.9GetModality	380
25.121.5.10GetModel	380
25.121.5.11GetNumberOfSeriesInStudy	380
25.121.5.12GetNumberOfStudyRelatedSeries	380
25.121.5.13GetPatientAge	380
25.121.5.14GetPatientDOB	380
25.121.5.15GetPatientID	380
25.121.5.16GetPatientName	381
25.121.5.17GetPatientSex	381
25.121.5.18GetScanOptions	381
25.121.5.19GetStudyDate	381
25.121.5.20GetStudyDescription	381
25.121.5.21GetStudyID	381
25.121.5.22GetValueFromTag	381
25.121.5.23InternalReadImageInformation	381
25.121.5.24BooleanMacro	381
25.121.5.25BooleanMacro	381
25.121.5.26BooleanMacro	381
25.121.5.27GetEnumMacro	381
25.121.5.28GetMacro	381
25.121.5.29GetMacro	381
25.121.5.30GetMacro	381
25.121.5.31GetMacro	381
25.121.5.32GetMacro	381
25.121.5.33GetStringMacro	381
25.121.5.34GetStringMacro	382
25.121.5.35GetStringMacro	382
25.121.5.36GetStringMacro	382
25.121.5.37NewMacro	382

25.121.5.38	SetEnumMacro	382
25.121.5.39	SetMacro	382
25.121.5.40	SetMacro	382
25.121.5.41	SetMacro	382
25.121.5.42	SetMacro	382
25.121.5.43	SetStringMacro	382
25.121.5.44	TypeMacro	382
25.121.5.45	LoadPrivateTagsDefaultOff	382
25.121.5.46	LoadPrivateTagsDefaultOn	382
25.121.5.47	LoadSequencesDefaultOff	382
25.121.5.48	LoadSequencesDefaultOn	383
25.121.5.49	OpenGDCMFileForReading	383
25.121.5.50	OpenGDCMFileForWriting	383
25.121.5.51	PrintSelf	383
25.121.5.52	Read	383
25.121.5.53	ReadImageInformation	383
25.121.5.54	SetLoadPrivateTagsDefault	383
25.121.5.55	SetLoadSequencesDefault	383
25.121.5.56	Write	383
25.121.5.57	WriteImageInformation	383
25.121.6	Member Data Documentation	383
25.121.6.1	m_FrameOfReferenceInstanceUID	383
25.121.6.2	m_KeepOriginalUID	383
25.121.6.3	m_RescaleIntercept	383
25.121.6.4	m_RescaleSlope	383
25.121.6.5	m_SeriesInstanceUID	383
25.121.6.6	m_StudyInstanceUID	384
25.121.6.7	m_UIDPrefix	384
25.122	dcm::Global Class Reference	384
25.122.1	Detailed Description	384
25.122.2	Constructor & Destructor Documentation	385
25.122.2.1	Global	385
25.122.2.2	~Global	385
25.122.3	Member Function Documentation	385
25.122.3.1	Append	385
25.122.3.2	GetDefs	385
25.122.3.3	GetDicts	385

25.122.3.4	GetDicts	385
25.122.3.5	GetInstance	385
25.122.3.6	LoadResourcesFiles	386
25.122.3.7	Locate	386
25.122.3.8	Prepend	386
25.122.4	Friends And Related Function Documentation	386
25.122.4.1	operator<<	386
25.123	dcm::GroupDict Class Reference	386
25.123.1	Detailed Description	387
25.123.2	Member Typedef Documentation	387
25.123.2.1	GroupStringVector	387
25.123.3	Constructor & Destructor Documentation	387
25.123.3.1	GroupDict	387
25.123.3.2	~GroupDict	387
25.123.4	Member Function Documentation	387
25.123.4.1	Add	387
25.123.4.2	GetAbbreviation	387
25.123.4.3	GetName	387
25.123.4.4	Insert	387
25.123.4.5	Size	387
25.123.5	Friends And Related Function Documentation	388
25.123.5.1	operator<<	388
25.124	dcm::IconImageFilter Class Reference	388
25.124.1	Detailed Description	388
25.124.2	Constructor & Destructor Documentation	389
25.124.2.1	IconImageFilter	389
25.124.2.2	~IconImageFilter	389
25.124.3	Member Function Documentation	389
25.124.3.1	Extract	389
25.124.3.2	ExtractIconImages	389
25.124.3.3	ExtractVeprolIconImages	389
25.124.3.4	GetFile	389
25.124.3.5	GetFile	389
25.124.3.6	GetIconImage	389
25.124.3.7	GetNumberOfIconImages	390
25.124.3.8	SetFile	390
25.125	dcm::IconImageGenerator Class Reference	390

25.125.1	Detailed Description	391
25.125.2	Constructor & Destructor Documentation	391
25.125.2.1	IconImageGenerator	391
25.125.2.2	~IconImageGenerator	391
25.125.3	Member Function Documentation	391
25.125.3.1	AutoPixelMinMax	391
25.125.3.2	ConvertRGBToPaletteColor	391
25.125.3.3	Generate	391
25.125.3.4	GetIconImage	392
25.125.3.5	GetPixmap	392
25.125.3.6	GetPixmap	392
25.125.3.7	SetOutputDimensions	392
25.125.3.8	SetOutsideValuePixel	392
25.125.3.9	SetPixelMinMax	392
25.125.3.10	SetPixmap	392
25.126	dcm::ignore_char Struct Reference	392
25.126.1	Constructor & Destructor Documentation	393
25.126.1.1	ignore_char	393
25.126.2	Member Data Documentation	393
25.126.2.1	m_char	393
25.127	dcm::Image Class Reference	393
25.127.1	Detailed Description	394
25.127.2	Constructor & Destructor Documentation	395
25.127.2.1	Image	395
25.127.2.2	~Image	395
25.127.3	Member Function Documentation	395
25.127.3.1	GetDirectionCosines	395
25.127.3.2	GetDirectionCosines	395
25.127.3.3	GetIntercept	395
25.127.3.4	GetOrigin	395
25.127.3.5	GetOrigin	396
25.127.3.6	GetSlope	396
25.127.3.7	GetSpacing	396
25.127.3.8	GetSpacing	396
25.127.3.9	Print	396
25.127.3.10	SetDirectionCosines	396
25.127.3.11	SetDirectionCosines	396

25.127.3.12	SetDirectionCosines	396
25.127.3.13	SetIntercept	396
25.127.3.13	SetOrigin	396
25.127.3.15	SetOrigin	396
25.127.3.16	SetOrigin	396
25.127.3.16	SetSlope	396
25.127.3.18	SetSpacing	396
25.127.3.19	SetSpacing	397
25.128	dcm::ImageApplyLookupTable Class Reference	397
25.128.1	Detailed Description	399
25.128.2	Constructor & Destructor Documentation	399
25.128.2.1	ImageApplyLookupTable	399
25.128.2.2	~ImageApplyLookupTable	399
25.128.3	Member Function Documentation	399
25.128.3.1	Apply	399
25.129	dcm::ImageChangePhotometricInterpretation Class Reference	399
25.129.1	Detailed Description	402
25.129.2	Constructor & Destructor Documentation	402
25.129.2.1	ImageChangePhotometricInterpretation	402
25.129.2.2	~ImageChangePhotometricInterpretation	402
25.129.3	Member Function Documentation	402
25.129.3.1	Change	402
25.129.3.2	ChangeMonochrome	402
25.129.3.3	GetPhotometricInterpretation	402
25.129.3.4	RGB2YBR	402
25.129.3.5	SetPhotometricInterpretation	402
25.129.3.6	YBR2RGB	402
25.130	dcm::ImageChangePlanarConfiguration Class Reference	403
25.130.1	Detailed Description	405
25.130.2	Constructor & Destructor Documentation	405
25.130.2.1	ImageChangePlanarConfiguration	405
25.130.2.2	~ImageChangePlanarConfiguration	405
25.130.3	Member Function Documentation	405
25.130.3.1	Change	405
25.130.3.2	GetPlanarConfiguration	405
25.130.3.3	RGBPixelsToRGBPlanes	405
25.130.3.4	RGBPlanesToRGBPixels	405

25.130.3.5SetPlanarConfiguration	405
25.131dcm::ImageChangeTransferSyntax Class Reference	406
25.131.1Detailed Description	408
25.131.2Constructor & Destructor Documentation	408
25.131.2.1ImageChangeTransferSyntax	408
25.131.2.2~ImageChangeTransferSyntax	408
25.131.3Member Function Documentation	408
25.131.3.1Change	408
25.131.3.2GetTransferSyntax	408
25.131.3.3SetCompressIconImage	409
25.131.3.4SetForce	409
25.131.3.5SetTransferSyntax	409
25.131.3.6SetUserCodec	409
25.131.3.7TryJPEG2000Codec	409
25.131.3.8TryJPEGCodec	409
25.131.3.9TryJPEGLSCodec	409
25.131.3.10TryRAWCodec	409
25.131.3.11TryRLECodec	409
25.132dcm::ImageCodec Class Reference	410
25.132.1Detailed Description	412
25.132.2Member Typedef Documentation	412
25.132.2.1LUTPtr	412
25.132.3Constructor & Destructor Documentation	412
25.132.3.1ImageCodec	412
25.132.3.2~ImageCodec	412
25.132.4Member Function Documentation	412
25.132.4.1CanCode	412
25.132.4.2CanDecode	412
25.132.4.3Decode	412
25.132.4.4Decode	413
25.132.4.5DoByteSwap	413
25.132.4.6DoInvertMonochrome	413
25.132.4.7DoOverlayCleanup	413
25.132.4.8DoPaddedCompositePixelCode	413
25.132.4.9DoPlanarConfiguration	413
25.132.4.10DoSimpleCopy	413
25.132.4.11DoYBR	413

25.132.4.10	GetDimensions	413
25.132.4.10	GetHeaderInfo	413
25.132.4.10	GetLossyFlag	413
25.132.4.10	GetLUT	413
25.132.4.10	GetNeedByteSwap	413
25.132.4.10	GetNumberOfDimensions	413
25.132.4.10	GetPhotometricInterpretation	413
25.132.4.10	GetPixelFormat	413
25.132.4.20	GetPixelFormat	413
25.132.4.20	GetPlanarConfiguration	413
25.132.4.20	Lossy	413
25.132.4.20	Valid	414
25.132.4.20	SetDimensions	414
25.132.4.20	SetDimensions	414
25.132.4.20	SetLossyFlag	414
25.132.4.20	SetLUT	414
25.132.4.20	SetNeedByteSwap	414
25.132.4.20	SetNeedOverlayCleanup	414
25.132.4.30	SetNumberOfDimensions	414
25.132.4.30	SetPhotometricInterpretation	414
25.132.4.30	SetPixelFormat	414
25.132.4.30	SetPlanarConfiguration	414
25.132.5	Friends And Related Function Documentation	414
25.132.5.1	ImageChangePhotometricInterpretation	414
25.132.6	Member Data Documentation	414
25.132.6.1	Dimensions	415
25.132.6.2	LossyFlag	415
25.132.6.3	LUT	415
25.132.6.4	NeedByteSwap	415
25.132.6.5	NeedOverlayCleanup	415
25.132.6.6	NumberOfDimensions	415
25.132.6.7	PF	415
25.132.6.8	PI	415
25.132.6.9	PlanarConfiguration	415
25.132.6.10	RequestPaddedCompositePixelCode	415
25.132.6.10	RequestPlanarConfiguration	415
25.132.6	gdcmm::ImageConverter Class Reference	415

25.133.1	Detailed Description	415
25.133.2	Constructor & Destructor Documentation	416
25.133.2.1	ImageConverter	416
25.133.2.2	~ImageConverter	416
25.133.3	Member Function Documentation	416
25.133.3.1	Convert	416
25.133.3.2	GetOutput	416
25.133.3.3	SetInput	416
25.134	dcm::ImageFragmentSplitter Class Reference	416
25.134.1	Detailed Description	418
25.134.2	Constructor & Destructor Documentation	418
25.134.2.1	ImageFragmentSplitter	418
25.134.2.2	~ImageFragmentSplitter	418
25.134.3	Member Function Documentation	418
25.134.3.1	GetFragmentSizeMax	418
25.134.3.2	SetForce	418
25.134.3.3	SetFragmentSizeMax	418
25.134.3.4	Split	418
25.135	dcm::ImageHelper Class Reference	418
25.135.1	Detailed Description	419
25.135.2	Member Function Documentation	419
25.135.2.1	ComputeSpacingFromImagePositionPatient	419
25.135.2.2	GetDimensionsValue	419
25.135.2.3	GetDirectionCosinesFromDataSet	420
25.135.2.4	GetDirectionCosinesValue	420
25.135.2.5	GetForcePixelSpacing	420
25.135.2.6	GetForceRescaleInterceptSlope	420
25.135.2.7	GetLUT	420
25.135.2.8	GetOriginValue	420
25.135.2.9	GetPhotometricInterpretationValue	420
25.135.2.10	GetPixelFormatValue	420
25.135.2.11	GetPlanarConfigurationValue	420
25.135.2.12	GetPointerFromElement	420
25.135.2.13	GetRescaleInterceptSlopeValue	420
25.135.2.14	GetSpacingTagFromMediaStorage	420
25.135.2.15	GetSpacingValue	420
25.135.2.16	GetZSpacingTagFromMediaStorage	421

25.135.2.1	SetDimensionsValue	421
25.135.2.1	SetDirectionCosinesValue	421
25.135.2.1	SetForcePixelSpacing	421
25.135.2.2	SetForceRescaleInterceptSlope	421
25.135.2.2	SetOriginValue	421
25.135.2.2	SetRescaleInterceptSlopeValue	421
25.135.2.2	SetSpacingValue	421
25.136	gdcm::ImageReader Class Reference	421
25.136.1	Detailed Description	424
25.136.2	Constructor & Destructor Documentation	424
25.136.2.1	ImageReader	424
25.136.2.2	~ImageReader	424
25.136.3	Member Function Documentation	424
25.136.3.1	GetImage	424
25.136.3.2	GetImage	424
25.136.3.3	Read	424
25.136.3.4	ReadACRNEMAIImage	425
25.136.3.5	ReadImage	425
25.137	gdcm::ImageToImageFilter Class Reference	425
25.137.1	Detailed Description	426
25.137.2	Constructor & Destructor Documentation	426
25.137.2.1	ImageToImageFilter	427
25.137.2.2	~ImageToImageFilter	427
25.137.3	Member Function Documentation	427
25.137.3.1	GetInput	427
25.137.3.2	GetOutput	427
25.138	gdcm::ImageWriter Class Reference	427
25.138.1	Detailed Description	429
25.138.2	Constructor & Destructor Documentation	429
25.138.2.1	ImageWriter	429
25.138.2.2	~ImageWriter	429
25.138.3	Member Function Documentation	429
25.138.3.1	GetImage	429
25.138.3.2	GetImage	429
25.138.3.3	Write	429
25.139	gdcm::network::ImplementationClassUIDSub Class Reference	430
25.139.1	Detailed Description	430

25.139.2	Constructor & Destructor Documentation	430
25.139.2.1	ImplementationClassUIDSub	430
25.139.3	Member Function Documentation	430
25.139.3.1	Read	430
25.139.3.2	Size	430
25.139.3.3	Write	430
25.140	dcm::network::ImplementationUIDSub Class Reference	430
25.140.1	Detailed Description	430
25.140.2	Constructor & Destructor Documentation	431
25.140.2.1	ImplementationUIDSub	431
25.140.3	Member Function Documentation	431
25.140.3.1	Write	431
25.141	dcm::network::ImplementationVersionNameSub Class Reference	431
25.141.1	Detailed Description	431
25.141.2	Constructor & Destructor Documentation	431
25.141.2.1	ImplementationVersionNameSub	431
25.141.3	Member Function Documentation	431
25.141.3.1	Read	431
25.141.3.2	Size	431
25.141.3.3	Write	431
25.142	dcm::ImplicitDataElement Class Reference	432
25.142.1	Detailed Description	433
25.142.2	Member Function Documentation	433
25.142.2.1	GetLength	433
25.142.2.2	Read	433
25.142.2.3	ReadPreValue	433
25.142.2.4	ReadValue	433
25.142.2.5	ReadWithLength	433
25.142.2.6	Write	433
25.143	dcm::InitializeEvent Class Reference	434
25.144	dcm::IOD Class Reference	435
25.144.1	Detailed Description	435
25.144.2	Member Typedef Documentation	435
25.144.2.1	MapIODEntry	435
25.144.2.2	SizeType	435
25.144.3	Constructor & Destructor Documentation	435
25.144.3.1	IOD	435

25.144.4	Member Function Documentation	436
25.144.4.1	AddIODEntry	436
25.144.4.2	Clear	436
25.144.4.3	GetIODEntry	436
25.144.4.4	GetNumberOfIODs	436
25.144.4.5	GetTypeFromTag	436
25.144.5	Friends And Related Function Documentation	436
25.144.5.1	operator<<	436
25.145	dcm::IODEntry Class Reference	436
25.145.1	Detailed Description	437
25.145.2	Constructor & Destructor Documentation	437
25.145.2.1	IODEntry	437
25.145.3	Member Function Documentation	437
25.145.3.1	GetIE	437
25.145.3.2	GetName	437
25.145.3.3	GetRef	437
25.145.3.4	GetUsage	438
25.145.3.5	GetUsageType	438
25.145.3.6	SetIE	438
25.145.3.7	SetName	438
25.145.3.8	SetRef	438
25.145.3.9	SetUsage	438
25.145.4	Friends And Related Function Documentation	438
25.145.4.1	operator<<	438
25.146	dcm::IODs Class Reference	438
25.146.1	Detailed Description	439
25.146.2	Member Typedef Documentation	439
25.146.2.1	IODMapType	439
25.146.2.2	IODMapTypeConstIterator	439
25.146.2.3	IODName	439
25.146.3	Constructor & Destructor Documentation	439
25.146.3.1	IODs	439
25.146.4	Member Function Documentation	439
25.146.4.1	AddIOD	439
25.146.4.2	Begin	439
25.146.4.3	Clear	439
25.146.4.4	End	439

25.146.4.5	GetIOD	439
25.146.5	Friends And Related Function Documentation	439
25.146.5.1	operator<<	439
25.147	dcm::IPPSorter Class Reference	439
25.147.1	Detailed Description	441
25.147.2	Constructor & Destructor Documentation	441
25.147.2.1	IPPSorter	441
25.147.2.2	~IPPSorter	441
25.147.3	Member Function Documentation	441
25.147.3.1	GetDirectionCosinesTolerance	441
25.147.3.2	GetZSpacing	441
25.147.3.3	GetZSpacingTolerance	442
25.147.3.4	SetComputeZSpacing	442
25.147.3.5	SetDirectionCosinesTolerance	442
25.147.3.6	SetZSpacingTolerance	442
25.147.3.7	Sort	442
25.147.4	Member Data Documentation	443
25.147.4.1	ComputeZSpacing	443
25.147.4.2	DirCosTolerance	443
25.147.4.3	ZSpacing	443
25.147.4.4	ZTolerance	443
25.148	dcm::Item Class Reference	443
25.148.1	Detailed Description	445
25.148.2	Constructor & Destructor Documentation	445
25.148.2.1	Item	445
25.148.2.2	Item	445
25.148.3	Member Function Documentation	445
25.148.3.1	Clear	445
25.148.3.2	FindDataElement	445
25.148.3.3	GetDataElement	445
25.148.3.4	GetLength	445
25.148.3.5	GetNestedDataSet	446
25.148.3.6	GetNestedDataSet	446
25.148.3.7	InsertDataElement	446
25.148.3.8	Read	446
25.148.3.9	SetNestedDataSet	446
25.148.3.10	Write	446

25.148.4	Friends And Related Function Documentation	446
25.148.4.1	operator<<	446
25.149	gdcm::IterationEvent Class Reference	446
25.150	gdcm::JPEG12Codec Class Reference	447
25.150.1	Detailed Description	449
25.150.2	Constructor & Destructor Documentation	449
25.150.2.1	JPEG12Codec	449
25.150.2.2	~JPEG12Codec	449
25.150.3	Member Function Documentation	449
25.150.3.1	Decode	449
25.150.3.2	GetHeaderInfo	449
25.150.3.3	InternalCode	449
25.151	gdcm::JPEG16Codec Class Reference	449
25.151.1	Detailed Description	451
25.151.2	Constructor & Destructor Documentation	451
25.151.2.1	JPEG16Codec	451
25.151.2.2	~JPEG16Codec	451
25.151.3	Member Function Documentation	451
25.151.3.1	Decode	451
25.151.3.2	GetHeaderInfo	451
25.151.3.3	InternalCode	451
25.152	gdcm::JPEG2000Codec Class Reference	451
25.152.1	Detailed Description	453
25.152.2	Constructor & Destructor Documentation	453
25.152.2.1	JPEG2000Codec	453
25.152.2.2	~JPEG2000Codec	453
25.152.3	Member Function Documentation	453
25.152.3.1	CanCode	453
25.152.3.2	CanDecode	453
25.152.3.3	Code	454
25.152.3.4	Decode	454
25.152.3.5	Decode	454
25.152.3.6	GetHeaderInfo	454
25.152.3.7	GetQuality	454
25.152.3.8	GetRate	454
25.152.3.9	SetNumberOfResolutions	454
25.152.3.10	SetQuality	454

25.152.3.1	SetRate	454
25.152.3.1	SetReversible	454
25.152.3.1	SetTileSize	454
25.152.4	Friends And Related Function Documentation	454
25.152.4.1	Bitmap	454
25.153	dcm::JPEG8Codec Class Reference	454
25.153.1	Detailed Description	456
25.153.2	Constructor & Destructor Documentation	456
25.153.2.1	JPEG8Codec	456
25.153.2.2	~JPEG8Codec	456
25.153.3	Member Function Documentation	456
25.153.3.1	Decode	456
25.153.3.2	GetHeaderInfo	456
25.153.3.3	InternalCode	456
25.154	dcm::JPEGCodec Class Reference	456
25.154.1	Detailed Description	458
25.154.2	Constructor & Destructor Documentation	458
25.154.2.1	JPEGCodec	458
25.154.2.2	~JPEGCodec	458
25.154.3	Member Function Documentation	459
25.154.3.1	CanCode	459
25.154.3.2	CanDecode	459
25.154.3.3	Code	459
25.154.3.4	ComputeOffsetTable	459
25.154.3.5	Decode	459
25.154.3.6	Decode	459
25.154.3.7	GetHeaderInfo	459
25.154.3.8	GetLossless	459
25.154.3.9	GetQuality	459
25.154.3.10	Valid	460
25.154.3.11	SetBitSample	460
25.154.3.12	SetLossless	460
25.154.3.13	SetPixelFormat	460
25.154.3.13	SetQuality	460
25.154.4	Member Data Documentation	460
25.154.4.1	BitSample	460
25.154.4.2	Lossless	460

25.154.4.3Quality	460
25.155gdcmm::JPEGLSCodec Class Reference	460
25.155.1Detailed Description	462
25.155.2Constructor & Destructor Documentation	462
25.155.2.1JPEGLSCodec	462
25.155.2.2~JPEGLSCodec	462
25.155.3Member Function Documentation	462
25.155.3.1CanCode	462
25.155.3.2CanDecode	462
25.155.3.3Code	462
25.155.3.4Decode	463
25.155.3.5Decode	463
25.155.3.6GetBufferLength	463
25.155.3.7GetHeaderInfo	463
25.155.3.8GetLossless	463
25.155.3.9SetBufferLength	463
25.155.3.10SetLossless	463
25.155.3.11SetLossyError	463
25.156gdcmm::KAKADUCodec Class Reference	463
25.156.1Detailed Description	465
25.156.2Constructor & Destructor Documentation	465
25.156.2.1KAKADUCodec	465
25.156.2.2~KAKADUCodec	465
25.156.3Member Function Documentation	465
25.156.3.1CanCode	465
25.156.3.2CanDecode	465
25.156.3.3Code	465
25.156.3.4Decode	465
25.157gdcmm::LO Class Reference	465
25.157.1Detailed Description	466
25.157.2Member Typedef Documentation	466
25.157.2.1const_iterator	466
25.157.2.2const_reference	466
25.157.2.3const_reverse_iterator	466
25.157.2.4difference_type	466
25.157.2.5iterator	466
25.157.2.6pointer	466

25.157.2.7reference	466
25.157.2.8reverse_iterator	466
25.157.2.9size_type	466
25.157.2.10Superclass	466
25.157.2.11Value_type	467
25.157.3Constructor & Destructor Documentation	467
25.157.3.1LO	467
25.157.3.2LO	467
25.157.3.3LO	467
25.157.3.4LO	467
25.157.4Member Function Documentation	467
25.157.4.1IsValid	467
25.158gdcmm::LookupTable Class Reference	467
25.158.1Detailed Description	469
25.158.2Member Enumeration Documentation	469
25.158.2.1LookupTableType	469
25.158.3Constructor & Destructor Documentation	469
25.158.3.1LookupTable	469
25.158.3.2~LookupTable	469
25.158.3.3LookupTable	469
25.158.4Member Function Documentation	469
25.158.4.1Allocate	470
25.158.4.2Clear	470
25.158.4.3Decode	470
25.158.4.4GetBitSample	470
25.158.4.5GetBufferAsRGBA	470
25.158.4.6GetLUT	470
25.158.4.7GetLUTDescriptor	470
25.158.4.8GetLUTLength	470
25.158.4.9GetPointer	470
25.158.4.10InitializeBlueLUT	470
25.158.4.11Initialized	470
25.158.4.12InitializeGreenLUT	470
25.158.4.13InitializeLUT	470
25.158.4.14InitializeRedLUT	470
25.158.4.15Print	471
25.158.4.16SetBlueLUT	471

25.158.4.1	SetGreenLUT	471
25.158.4.1	SetLUT	471
25.158.4.1	SetRedLUT	471
25.158.4.2	WriteBufferAsRGBA	471
25.158.5	Member Data Documentation	471
25.158.5.1	BitSample	471
25.158.5.2	IncompleteLUT	471
25.158.5.3	Internal	471
25.159	dcm::Scanner::Itstr Struct Reference	471
25.159.1	Member Function Documentation	471
25.159.1.1	operator()	471
25.160	dcm::Macro Class Reference	472
25.160.1	Detailed Description	472
25.160.2	Member Typedef Documentation	472
25.160.2.1	ArrayIncludeMacrosType	472
25.160.2.2	MapModuleEntry	472
25.160.3	Constructor & Destructor Documentation	472
25.160.3.1	Macro	472
25.160.4	Member Function Documentation	473
25.160.4.1	AddMacroEntry	473
25.160.4.2	Clear	473
25.160.4.3	FindMacroEntry	473
25.160.4.4	GetMacroEntry	473
25.160.4.5	GetName	473
25.160.4.6	SetName	473
25.160.4.7	Verify	473
25.160.5	Friends And Related Function Documentation	473
25.160.5.1	operator<<	473
25.161	dcm::Macros Class Reference	473
25.161.1	Detailed Description	474
25.161.2	Member Typedef Documentation	474
25.161.2.1	ModuleMapType	474
25.161.3	Constructor & Destructor Documentation	474
25.161.3.1	Macros	474
25.161.4	Member Function Documentation	474
25.161.4.1	AddMacro	474
25.161.4.2	Clear	474

25.161.4.3	GetMacro	474
25.161.4.4	IsEmpty	474
25.161.5	Friends And Related Function Documentation	474
25.161.5.1	operator<<	474
25.162	dcm::network::MaximumLengthSub Class Reference	474
25.162.1	Detailed Description	475
25.162.2	Constructor & Destructor Documentation	475
25.162.2.1	MaximumLengthSub	475
25.162.3	Member Function Documentation	475
25.162.3.1	GetMaximumLength	475
25.162.3.2	Read	475
25.162.3.3	SetMaximumLength	475
25.162.3.4	Size	475
25.162.3.5	Write	475
25.163	dcm::MD5 Class Reference	475
25.163.1	Detailed Description	476
25.163.2	Constructor & Destructor Documentation	476
25.163.2.1	MD5	476
25.163.2.2	~MD5	476
25.163.3	Member Function Documentation	476
25.163.3.1	Compute	476
25.163.3.2	ComputeFile	476
25.164	dcm::MediaStorage Class Reference	476
25.164.1	Detailed Description	479
25.164.2	Member Enumeration Documentation	479
25.164.2.1	MSType	479
25.164.2.2	ObjectType	481
25.164.3	Constructor & Destructor Documentation	481
25.164.3.1	MediaStorage	481
25.164.4	Member Function Documentation	481
25.164.4.1	GetModality	481
25.164.4.2	GetModalityDimension	481
25.164.4.3	GetMSString	481
25.164.4.4	GetMSType	482
25.164.4.5	GetNumberOfModality	482
25.164.4.6	GetNumberOfMSString	482
25.164.4.7	GetNumberOfMSType	482

25.164.4.8	GetString	482
25.164.4.9	GuessFromModality	482
25.164.4.10	Image	482
25.164.4.11	Undefined	482
25.164.4.12	operator MType	482
25.164.4.13	SetFromDataSet	482
25.164.4.14	SetFromFile	482
25.164.4.15	SetFromHeader	483
25.164.4.16	SetFromModality	483
25.164.4.17	SetFromSourceImageSequence	483
25.164.5	Friends And Related Function Documentation	483
25.164.5.1	operator<<	483
25.165	dcm::MemberCommand< T > Class Template Reference	483
25.165.1	Detailed Description	485
25.165.2	Member Typedef Documentation	485
25.165.2.1	Self	485
25.165.2.2	TConstMemberFunctionPointer	485
25.165.2.3	TMemberFunctionPointer	485
25.165.3	Constructor & Destructor Documentation	485
25.165.3.1	MemberCommand	485
25.165.3.2	~MemberCommand	485
25.165.4	Member Function Documentation	485
25.165.4.1	Execute	485
25.165.4.2	Execute	486
25.165.4.3	New	486
25.165.4.4	SetCallbackFunction	486
25.165.4.5	SetCallbackFunction	486
25.165.5	Member Data Documentation	486
25.165.5.1	m_ConstMemberFunction	486
25.165.5.2	m_MemberFunction	486
25.165.5.3	m_This	486
25.166	dcm::MeshPrimitive Class Reference	487
25.166.1	Detailed Description	488
25.166.2	Member Typedef Documentation	489
25.166.2.1	PrimitivesData	489
25.166.3	Member Enumeration Documentation	489
25.166.3.1	MPType	489

25.166.4	Constructor & Destructor Documentation	489
25.166.4.1	MeshPrimitive	489
25.166.4.2	~MeshPrimitive	489
25.166.5	Member Function Documentation	489
25.166.5.1	AddPrimitiveData	489
25.166.5.2	GetMPType	489
25.166.5.3	GetMPTypeString	489
25.166.5.4	GetNumberOfPrimitivesData	489
25.166.5.5	GetPrimitiveData	489
25.166.5.6	GetPrimitiveData	489
25.166.5.7	GetPrimitiveData	489
25.166.5.8	GetPrimitiveData	489
25.166.5.9	GetPrimitivesData	489
25.166.5.10	GetPrimitivesData	490
25.166.5.11	GetPrimitiveType	490
25.166.5.12	SetPrimitiveData	490
25.166.5.13	SetPrimitiveData	490
25.166.5.14	SetPrimitivesData	490
25.166.5.15	SetPrimitiveType	490
25.166.6	Member Data Documentation	490
25.166.6.1	PrimitiveData	490
25.166.6.2	PrimitiveType	490
25.167	dcm::ModifiedEvent Class Reference	490
25.168	dcm::Module Class Reference	491
25.168.1	Detailed Description	492
25.168.2	Member Typedef Documentation	492
25.168.2.1	ArrayIncludeMacrosType	492
25.168.2.2	MapModuleEntry	492
25.168.3	Constructor & Destructor Documentation	492
25.168.3.1	Module	492
25.168.4	Member Function Documentation	492
25.168.4.1	AddMacro	492
25.168.4.2	AddModuleEntry	492
25.168.4.3	Clear	492
25.168.4.4	FindModuleEntryInMacros	492
25.168.4.5	GetModuleEntryInMacros	492
25.168.4.6	GetName	493

25.168.4.7SetName	493
25.168.4.8Verify	493
25.168.5Friends And Related Function Documentation	493
25.168.5.1operator<<	493
25.169dcm::ModuleEntry Class Reference	493
25.169.1Detailed Description	495
25.169.2Member Typedef Documentation	495
25.169.2.1Description	495
25.169.3Constructor & Destructor Documentation	495
25.169.3.1ModuleEntry	495
25.169.3.2~ModuleEntry	495
25.169.4Member Function Documentation	495
25.169.4.1GetDescription	495
25.169.4.2GetName	495
25.169.4.3GetType	495
25.169.4.4SetDescription	495
25.169.4.5SetName	495
25.169.4.6SetType	495
25.169.5Friends And Related Function Documentation	495
25.169.5.1operator<<	495
25.169.6Member Data Documentation	496
25.169.6.1DataElementType	496
25.169.6.2DescriptionField	496
25.169.6.3Name	496
25.170dcm::Modules Class Reference	496
25.170.1Detailed Description	496
25.170.2Member Typedef Documentation	497
25.170.2.1ModuleMapType	497
25.170.3Constructor & Destructor Documentation	497
25.170.3.1Modules	497
25.170.4Member Function Documentation	497
25.170.4.1AddModule	497
25.170.4.2Clear	497
25.170.4.3GetModule	497
25.170.4.4IsEmpty	497
25.170.5Friends And Related Function Documentation	497
25.170.5.1operator<<	497

25.171	dcm::MovePatientRootQuery Class Reference	497
25.171.1	Detailed Description	499
25.171.2	Constructor & Destructor Documentation	499
25.171.2.1	MovePatientRootQuery	499
25.171.3	Member Function Documentation	499
25.171.3.1	GetAbstractSyntaxUID	499
25.171.3.2	GetTagListByLevel	499
25.171.3.3	InitializeDataSet	499
25.171.3.4	ValidateQuery	499
25.171.4	Friends And Related Function Documentation	499
25.171.4.1	QueryFactory	499
25.172	dcm::MoveStudyRootQuery Class Reference	500
25.172.1	Detailed Description	501
25.172.2	Constructor & Destructor Documentation	501
25.172.2.1	MoveStudyRootQuery	501
25.172.3	Member Function Documentation	501
25.172.3.1	GetAbstractSyntaxUID	501
25.172.3.2	GetTagListByLevel	501
25.172.3.3	InitializeDataSet	501
25.172.3.4	ValidateQuery	501
25.172.4	Friends And Related Function Documentation	502
25.172.4.1	QueryFactory	502
25.173	dcm::NestedModuleEntries Class Reference	502
25.173.1	Detailed Description	504
25.173.2	Member Typedef Documentation	504
25.173.2.1	SizeType	504
25.173.3	Constructor & Destructor Documentation	504
25.173.3.1	NestedModuleEntries	504
25.173.4	Member Function Documentation	504
25.173.4.1	AddModuleEntry	504
25.173.4.2	GetModuleEntry	504
25.173.4.3	GetModuleEntry	504
25.173.4.4	GetNumberOfModuleEntries	504
25.173.5	Friends And Related Function Documentation	504
25.173.5.1	operator<<	504
25.174	dcm::NoEvent Class Reference	504
25.174.1	Detailed Description	505

25.175	dcm::Object Class Reference	505
25.175.1	Detailed Description	507
25.175.2	Constructor & Destructor Documentation	507
25.175.2.1	Object	507
25.175.2.2	~Object	507
25.175.2.3	Object	507
25.175.3	Member Function Documentation	507
25.175.3.1	operator=	507
25.175.3.2	Print	507
25.175.3.3	Register	507
25.175.3.4	UnRegister	507
25.175.4	Friends And Related Function Documentation	507
25.175.4.1	operator<<	507
25.175.4.2	SmartPointer	507
25.176	dcm::OneShotReadBuf Struct Reference	508
25.176.1	Constructor & Destructor Documentation	508
25.176.1.1	OneShotReadBuf	508
25.177	dcm::Orientation Class Reference	508
25.177.1	Detailed Description	509
25.177.2	Member Enumeration Documentation	509
25.177.2.1	OrientationType	509
25.177.3	Constructor & Destructor Documentation	509
25.177.3.1	Orientation	509
25.177.3.2	~Orientation	509
25.177.4	Member Function Documentation	509
25.177.4.1	GetLabel	509
25.177.4.2	GetMajorAxisFromPatientRelativeDirectionCosine	509
25.177.4.3	GetObliquityThresholdCosineValue	509
25.177.4.4	GetType	509
25.177.4.5	Print	510
25.177.4.6	SetObliquityThresholdCosineValue	510
25.177.5	Friends And Related Function Documentation	510
25.177.5.1	operator<<	510
25.178	dcm::Overlay Class Reference	510
25.178.1	Detailed Description	512
25.178.2	Constructor & Destructor Documentation	513
25.178.2.1	Overlay	513

25.178.2.2~Overlay	513
25.178.2.3Overlay	513
25.178.3Member Function Documentation	513
25.178.3.1Decode	513
25.178.3.2Decompress	513
25.178.3.3GetBitPosition	513
25.178.3.4GetBitsAllocated	513
25.178.3.5GetBuffer	513
25.178.3.6GetColumns	513
25.178.3.7GetDescription	513
25.178.3.8GetGroup	513
25.178.3.9GetOrigin	513
25.178.3.10GetOverlayData	513
25.178.3.11GetRows	513
25.178.3.12GetType	514
25.178.3.13GetUnpackBuffer	514
25.178.3.14GrabOverlayFromPixelData	514
25.178.3.15Empty	514
25.178.3.16InPixelData	514
25.178.3.17InPixelData	514
25.178.3.18Zero	514
25.178.3.19Print	514
25.178.3.20SetBitPosition	514
25.178.3.21SetBitsAllocated	514
25.178.3.22SetColumns	514
25.178.3.23SetDescription	514
25.178.3.24SetFrameOrigin	514
25.178.3.25SetGroup	514
25.178.3.26SetNumberOfFrames	515
25.178.3.27SetOrigin	515
25.178.3.28SetOverlay	515
25.178.3.29SetRows	515
25.178.3.30SetType	515
25.178.3.31Update	515
25.179gdcmm::ParseException Class Reference	515
25.179.1Detailed Description	517
25.179.2Constructor & Destructor Documentation	517

25.179.2.1	ParseException	517
25.179.2.2	~ParseException	517
25.179.3	Member Function Documentation	517
25.179.3.1	GetLastElement	517
25.179.3.2	operator=	517
25.179.3.3	SetLastElement	517
25.180	dcm::Parser Class Reference	517
25.180.1	Detailed Description	518
25.180.2	Member Typedef Documentation	518
25.180.2.1	EndElementHandler	518
25.180.2.2	StartElementHandler	518
25.180.3	Member Enumeration Documentation	518
25.180.3.1	ErrorType	518
25.180.4	Constructor & Destructor Documentation	519
25.180.4.1	Parser	519
25.180.4.2	~Parser	519
25.180.5	Member Function Documentation	519
25.180.5.1	GetBuffer	519
25.180.5.2	GetCurrentByteIndex	519
25.180.5.3	GetErrorCode	519
25.180.5.4	GetErrorString	519
25.180.5.5	GetUserData	519
25.180.5.6	Parse	519
25.180.5.7	ParseBuffer	519
25.180.5.8	Process	519
25.180.5.9	SetElementHandler	519
25.180.5.10	SetUserData	519
25.181	dcm::Patient Class Reference	519
25.181.1	Detailed Description	519
25.181.2	Constructor & Destructor Documentation	520
25.181.2.1	Patient	520
25.182	dcm::network::PDataTFPDU Class Reference	520
25.182.1	Detailed Description	521
25.182.2	Member Typedef Documentation	521
25.182.2.1	SizeType	521
25.182.3	Constructor & Destructor Documentation	521
25.182.3.1	PDataTFPDU	521

25.182.4	Member Function Documentation	521
25.182.4.1	AddPresentationDataValue	521
25.182.4.2	GetNumberOfPresentationDataValues	521
25.182.4.3	GetPresentationDataValue	521
25.182.4.4	IsLastFragment	521
25.182.4.5	Print	521
25.182.4.6	Read	522
25.182.4.7	ReadInto	522
25.182.4.8	Size	522
25.182.4.9	Write	522
25.183	dcm::PDBelement Class Reference	522
25.183.1	Detailed Description	523
25.183.2	Constructor & Destructor Documentation	523
25.183.2.1	PDBelement	523
25.183.3	Member Function Documentation	523
25.183.3.1	GetName	523
25.183.3.2	GetValue	523
25.183.3.3	operator==	523
25.183.3.4	SetName	524
25.183.3.5	SetValue	524
25.183.4	Friends And Related Function Documentation	524
25.183.4.1	operator<<	524
25.183.5	Member Data Documentation	524
25.183.5.1	NameField	524
25.183.5.2	ValueField	524
25.184	dcm::PDBHeader Class Reference	524
25.184.1	Detailed Description	525
25.184.2	Constructor & Destructor Documentation	525
25.184.2.1	PDBHeader	525
25.184.2.2	~PDBHeader	525
25.184.3	Member Function Documentation	525
25.184.3.1	FindPDBelementByName	525
25.184.3.2	GetPDBeEnd	525
25.184.3.3	GetPDBelementByName	525
25.184.3.4	GetPDBInfoTag	525
25.184.3.5	LoadFromDataElement	525
25.184.3.6	Print	526

25.184.4	Friends And Related Function Documentation	526
25.184.4.1	operator<<	526
25.185	dcm::PDFCodec Class Reference	526
25.185.1	Detailed Description	527
25.185.2	Constructor & Destructor Documentation	527
25.185.2.1	PDFCodec	527
25.185.2.2	~PDFCodec	527
25.185.3	Member Function Documentation	527
25.185.3.1	CanCode	527
25.185.3.2	CanDecode	528
25.185.3.3	Decode	528
25.186	dcm::network::PDUFactory Class Reference	528
25.186.1	Detailed Description	528
25.186.2	Member Function Documentation	528
25.186.2.1	ConstructAbortPDU	528
25.186.2.2	ConstructPDU	528
25.186.2.3	ConstructReleasePDU	529
25.186.2.4	CreateCEchoPDU	529
25.186.2.5	CreateCFindPDU	529
25.186.2.6	CreateCMovePDU	529
25.186.2.7	CreateCStoreRQPDU	529
25.186.2.8	CreateCStoreRSPPDU	529
25.186.2.9	DetermineEventByPDU	529
25.186.2.10	GetPDVs	529
25.187	dcm::PersonName Class Reference	529
25.187.1	Detailed Description	530
25.187.2	Member Function Documentation	530
25.187.2.1	GetMaxLength	530
25.187.2.2	GetNumberOfComponents	530
25.187.2.3	Print	530
25.187.2.4	SetBlob	530
25.187.2.5	SetComponents	530
25.187.2.6	SetComponents	530
25.187.3	Member Data Documentation	530
25.187.3.1	Component	530
25.187.3.2	MaxLength	530
25.187.3.3	MaxNumberOfComponents	530

25.187.3.4	Padding	530
25.187.3.5	Separator	530
25.188	dcm::PhotometricInterpretation Class Reference	530
25.188.1	Detailed Description	531
25.188.2	Member Enumeration Documentation	532
25.188.2.1	PType	532
25.188.3	Constructor & Destructor Documentation	532
25.188.3.1	PhotometricInterpretation	532
25.188.4	Member Function Documentation	532
25.188.4.1	GetPString	532
25.188.4.2	GetPType	532
25.188.4.3	GetSamplesPerPixel	532
25.188.4.4	GetString	532
25.188.4.5	GetType	532
25.188.4.6	IsLossless	532
25.188.4.7	IsLossy	532
25.188.4.8	IsRetired	532
25.188.4.9	IsSameColorSpace	533
25.188.4.10	operator PType	533
25.188.5	Friends And Related Function Documentation	533
25.188.5.1	operator<<	533
25.189	dcm::PixelFormat Class Reference	533
25.189.1	Detailed Description	534
25.189.2	Member Enumeration Documentation	535
25.189.2.1	ScalarType	535
25.189.3	Constructor & Destructor Documentation	535
25.189.3.1	PixelFormat	535
25.189.3.2	PixelFormat	535
25.189.3.3	~PixelFormat	535
25.189.4	Member Function Documentation	535
25.189.4.1	GetBitsAllocated	535
25.189.4.2	GetBitsStored	535
25.189.4.3	GetHighBit	536
25.189.4.4	GetMax	536
25.189.4.5	GetMin	536
25.189.4.6	GetPixelRepresentation	536
25.189.4.7	GetPixelSize	536

25.189.4.8	GetSamplesPerPixel	536
25.189.4.9	GetScalarType	536
25.189.4.10	GetScalarTypeAsString	536
25.189.4.11	IsValid	536
25.189.4.12	Operator ScalarType	537
25.189.4.13	Operator!=	537
25.189.4.14	Operator!=	537
25.189.4.15	Operator==	537
25.189.4.16	Operator==	537
25.189.4.17	Print	537
25.189.4.18	SetBitsAllocated	537
25.189.4.19	SetBitsStored	537
25.189.4.20	SetHighBit	537
25.189.4.21	SetPixelRepresentation	537
25.189.4.22	SetSamplesPerPixel	537
25.189.4.23	SetScalarType	537
25.189.4.24	Validate	537
25.189.5	Friends And Related Function Documentation	537
25.189.5.1	Bitmap	537
25.189.5.2	Operator<<	538
25.190	gdcm::Pixmap Class Reference	538
25.190.1	Detailed Description	540
25.190.2	Constructor & Destructor Documentation	540
25.190.2.1	Pixmap	540
25.190.2.2	~Pixmap	540
25.190.3	Member Function Documentation	540
25.190.3.1	AreOverlaysInPixelData	540
25.190.3.2	GetCurve	540
25.190.3.3	GetCurve	540
25.190.3.4	GetIconImage	540
25.190.3.5	GetIconImage	540
25.190.3.6	GetNumberOfCurves	540
25.190.3.7	GetNumberOfOverlays	540
25.190.3.8	GetOverlay	540
25.190.3.9	GetOverlay	540
25.190.3.10	Print	540
25.190.3.11	RemoveOverlay	541

25.190.3.1 2 SetIconImage	541
25.190.3.1 3 SetNumberOfCurves	541
25.190.3.1 3 SetNumberOfOverlays	541
25.190.4Member Data Documentation	541
25.190.4.1Curves	541
25.190.4.2Icon	541
25.190.4.3Overlays	541
25.191 0 dcm::PixmapReader Class Reference	541
25.191.1Detailed Description	543
25.191.2Constructor & Destructor Documentation	543
25.191.2.1PixmapReader	543
25.191.2.2~PixmapReader	543
25.191.3Member Function Documentation	543
25.191.3.1GetPixmap	543
25.191.3.2GetPixmap	543
25.191.3.3Read	543
25.191.3.4ReadACRNEMAIImage	543
25.191.3.5ReadImage	543
25.191.4Member Data Documentation	544
25.191.4.1PixelData	544
25.192 0 dcm::PixmapToPixmapFilter Class Reference	544
25.192.1Detailed Description	545
25.192.2Constructor & Destructor Documentation	545
25.192.2.1PixmapToPixmapFilter	545
25.192.2.2~PixmapToPixmapFilter	546
25.192.3Member Function Documentation	546
25.192.3.1GetInput	546
25.192.3.2GetOutput	546
25.193 0 dcm::PixmapWriter Class Reference	546
25.193.1Detailed Description	548
25.193.2Constructor & Destructor Documentation	548
25.193.2.1PixmapWriter	548
25.193.2.2~PixmapWriter	548
25.193.3Member Function Documentation	548
25.193.3.1DoIconImage	548
25.193.3.2GetImage	548
25.193.3.3GetImage	548

25.193.3.4	GetPixmap	549
25.193.3.5	GetPixmap	549
25.193.3.6	PrepareWrite	549
25.193.3.7	SetImage	549
25.193.3.8	SetPixmap	549
25.193.3.9	Write	549
25.193.4	Member Data Documentation	549
25.193.4.1	PixelData	549
25.194	dcm::PNMCodec Class Reference	549
25.194.1	Detailed Description	551
25.194.2	Constructor & Destructor Documentation	551
25.194.2.1	PNMCodec	551
25.194.2.2	~PNMCodec	551
25.194.3	Member Function Documentation	551
25.194.3.1	CanCode	551
25.194.3.2	CanDecode	551
25.194.3.3	GetBufferLength	551
25.194.3.4	GetHeaderInfo	551
25.194.3.5	Read	551
25.194.3.6	SetBufferLength	551
25.194.3.7	Write	551
25.195	dcm::Preamble Class Reference	552
25.195.1	Detailed Description	552
25.195.2	Constructor & Destructor Documentation	552
25.195.2.1	Preamble	552
25.195.2.2	~Preamble	552
25.195.2.3	Preamble	552
25.195.3	Member Function Documentation	553
25.195.3.1	Clear	553
25.195.3.2	Create	553
25.195.3.3	GetInternal	553
25.195.3.4	GetLength	553
25.195.3.5	IsEmpty	553
25.195.3.6	IsValid	553
25.195.3.7	operator=	553
25.195.3.8	Print	553
25.195.3.9	Read	553

25.195.3.10 Remove	553
25.195.3.11 Valid	553
25.195.3.12 Write	553
25.195.4 Friends And Related Function Documentation	553
25.195.4.1 operator<<	553
25.196 dcm::PresentationContext Class Reference	553
25.196.1 Detailed Description	554
25.196.2 Member Typedef Documentation	554
25.196.2.1 SizeType	554
25.196.2.2 TransferSyntaxArrayType	554
25.196.3 Constructor & Destructor Documentation	554
25.196.3.1 PresentationContext	554
25.196.3.2 PresentationContext	554
25.196.4 Member Function Documentation	554
25.196.4.1 AddTransferSyntax	554
25.196.4.2 GetAbstractSyntax	554
25.196.4.3 GetNumberOfTransferSyntaxes	554
25.196.4.4 GetPresentationContextID	554
25.196.4.5 GetTransferSyntax	554
25.196.4.6 operator==	554
25.196.4.7 Print	555
25.196.4.8 SetAbstractSyntax	555
25.196.4.9 SetPresentationContextID	555
25.197 dcm::network::PresentationContextAC Class Reference	555
25.197.1 Detailed Description	555
25.197.2 Constructor & Destructor Documentation	555
25.197.2.1 PresentationContextAC	555
25.197.3 Member Function Documentation	555
25.197.3.1 GetPresentationContextID	555
25.197.3.2 GetTransferSyntax	555
25.197.3.3 Print	555
25.197.3.4 Read	555
25.197.3.5 SetPresentationContextID	556
25.197.3.6 SetTransferSyntax	556
25.197.3.7 Size	556
25.197.3.8 Write	556
25.198 dcm::PresentationContextGenerator Class Reference	556

25.198.1	Detailed Description	557
25.198.2	Member Typedef Documentation	557
25.198.2.1	PresentationContextArrayType	557
25.198.2.2	SizeType	557
25.198.3	Constructor & Destructor Documentation	557
25.198.3.1	PresentationContextGenerator	557
25.198.4	Member Function Documentation	557
25.198.4.1	AddPresentationContext	557
25.198.4.2	GenerateFromFilenames	557
25.198.4.3	GenerateFromUID	557
25.198.4.4	GetDefaultTransferSyntax	558
25.198.4.5	GetPresentationContexts	558
25.198.4.6	SetDefaultTransferSyntax	558
25.198.4.7	SetMergeModeToAbstractSyntax	558
25.198.4.8	SetMergeModeToTransferSyntax	558
25.199	dcm::network::PresentationContextRQ Class Reference	558
25.199.1	Detailed Description	559
25.199.2	Member Typedef Documentation	559
25.199.2.1	SizeType	559
25.199.3	Constructor & Destructor Documentation	559
25.199.3.1	PresentationContextRQ	559
25.199.3.2	PresentationContextRQ	559
25.199.3.3	PresentationContextRQ	559
25.199.4	Member Function Documentation	559
25.199.4.1	AddTransferSyntax	559
25.199.4.2	GetAbstractSyntax	559
25.199.4.3	GetAbstractSyntax	559
25.199.4.4	GetNumberOfTransferSyntaxes	559
25.199.4.5	GetPresentationContextID	559
25.199.4.6	GetTransferSyntax	559
25.199.4.7	GetTransferSyntax	559
25.199.4.8	GetTransferSyntaxes	560
25.199.4.9	operator==	560
25.199.4.10	Print	560
25.199.4.11	Read	560
25.199.4.12	SetAbstractSyntax	560
25.199.4.13	SetPresentationContextID	560

25.199.4.13	Size	560
25.199.4.14	Write	560
25.200	dcm::network::PresentationDataValue Class Reference	560
25.200.1	Detailed Description	561
25.200.2	Constructor & Destructor Documentation	561
25.200.2.1	PresentationDataValue	561
25.200.3	Member Function Documentation	561
25.200.3.1	ConcatenatePDVBlobs	561
25.200.3.2	GetBlob	561
25.200.3.3	GetIsCommand	561
25.200.3.4	GetIsLastFragment	561
25.200.3.5	GetMessageHeader	561
25.200.3.6	GetPresentationContextID	561
25.200.3.7	Print	561
25.200.3.8	Read	561
25.200.3.9	ReadInto	561
25.200.3.10	SetBlob	561
25.200.3.11	SetCommand	561
25.200.3.12	SetDataSet	561
25.200.3.13	SetLastFragment	561
25.200.3.14	SetMessageHeader	561
25.200.3.15	SetPresentationContextID	562
25.200.3.16	Size	562
25.200.3.17	Write	562
25.201	dcm::Printer Class Reference	562
25.201.1	Detailed Description	564
25.201.2	Member Enumeration Documentation	564
25.201.2.1	PrintStyles	564
25.201.3	Constructor & Destructor Documentation	564
25.201.3.1	Printer	564
25.201.3.2	~Printer	564
25.201.4	Member Function Documentation	564
25.201.4.1	GetPrintStyle	564
25.201.4.2	Print	564
25.201.4.3	PrintDataElement	564
25.201.4.4	PrintDataSet	564
25.201.4.5	PrintSQ	565

25.201.4.6	SetColor	565
25.201.4.7	SetFile	565
25.201.4.8	SetStyle	565
25.201.5	Member Data Documentation	565
25.201.5.1	F	565
25.201.5.2	MaxPrintLength	565
25.201.5.3	PrintStyle	565
25.202	dcm::PrivateDict Class Reference	565
25.202.1	Detailed Description	566
25.202.2	Constructor & Destructor Documentation	566
25.202.2.1	PrivateDict	566
25.202.2.2	~PrivateDict	566
25.202.3	Member Function Documentation	566
25.202.3.1	AddDictEntry	566
25.202.3.2	FindDictEntry	566
25.202.3.3	GetDictEntry	566
25.202.3.4	IsEmpty	566
25.202.3.5	LoadDefault	566
25.202.3.6	PrintXML	566
25.202.3.7	RemoveDictEntry	566
25.202.4	Friends And Related Function Documentation	566
25.202.4.1	Dicts	566
25.202.4.2	operator<<	566
25.203	dcm::PrivateTag Class Reference	567
25.203.1	Detailed Description	568
25.203.2	Constructor & Destructor Documentation	568
25.203.2.1	PrivateTag	568
25.203.3	Member Function Documentation	568
25.203.3.1	GetOwner	568
25.203.3.2	operator<	568
25.203.3.3	ReadFromCommaSeparatedString	568
25.203.3.4	SetOwner	568
25.203.4	Friends And Related Function Documentation	568
25.203.4.1	operator<<	568
25.204	dcm::ProgressEvent Class Reference	569
25.204.1	Detailed Description	570
25.204.2	Member Typedef Documentation	570

25.204.2.1Self	570
25.204.2.2Superclass	570
25.204.3Constructor & Destructor Documentation	570
25.204.3.1ProgressEvent	570
25.204.3.2~ProgressEvent	570
25.204.3.3ProgressEvent	570
25.204.4Member Function Documentation	570
25.204.4.1CheckEvent	570
25.204.4.2GetEventName	570
25.204.4.3GetProgress	570
25.204.4.4MakeObject	570
25.204.4.5SetProgress	571
25.205dcm::PVRGCodec Class Reference	571
25.205.1Detailed Description	572
25.205.2Constructor & Destructor Documentation	572
25.205.2.1PVRGCodec	572
25.205.2.2~PVRGCodec	572
25.205.3Member Function Documentation	572
25.205.3.1CanCode	572
25.205.3.2CanDecode	572
25.205.3.3Code	573
25.205.3.4Decode	573
25.206dcm::PythonFilter Class Reference	573
25.206.1Detailed Description	573
25.206.2Constructor & Destructor Documentation	573
25.206.2.1PythonFilter	573
25.206.2.2~PythonFilter	573
25.206.3Member Function Documentation	573
25.206.3.1GetFile	573
25.206.3.2GetFile	574
25.206.3.3SetDicts	574
25.206.3.4SetFile	574
25.206.3.5ToPyObject	574
25.206.3.6UseDictAlways	574
25.207dcm::QueryBase Class Reference	574
25.207.1Detailed Description	574
25.207.2Constructor & Destructor Documentation	575

25.207.2.1~QueryBase	575
25.207.3Member Function Documentation	575
25.207.3.1GetAllTags	575
25.207.3.2GetName	575
25.207.3.3GetOptionalTags	575
25.207.3.4GetQueryLevel	575
25.207.3.5GetRequiredTags	575
25.207.3.6GetUniqueTags	575
25.208dcm::QueryFactory Class Reference	576
25.208.1Detailed Description	576
25.208.2Member Function Documentation	576
25.208.2.1GetCharacterFromCurrentLocale	576
25.208.2.2ListCharSets	576
25.208.2.3ProduceCharacterSetDataElement	576
25.208.2.4ProduceQuery	577
25.209dcm::QueryImage Class Reference	577
25.209.1Detailed Description	578
25.209.2Member Function Documentation	578
25.209.2.1GetName	578
25.209.2.2GetOptionalTags	578
25.209.2.3GetQueryLevel	578
25.209.2.4GetRequiredTags	578
25.209.2.5GetUniqueTags	578
25.210dcm::QueryPatient Class Reference	578
25.210.1Detailed Description	579
25.210.2Member Function Documentation	579
25.210.2.1GetName	579
25.210.2.2GetOptionalTags	580
25.210.2.3GetQueryLevel	580
25.210.2.4GetRequiredTags	580
25.210.2.5GetUniqueTags	580
25.211dcm::QuerySeries Class Reference	580
25.211.1Detailed Description	581
25.211.2Member Function Documentation	581
25.211.2.1GetName	581
25.211.2.2GetOptionalTags	581
25.211.2.3GetQueryLevel	581

25.211.2.4	GetRequiredTags	581
25.211.2.5	GetUniqueTags	582
25.211.3	dcm::QueryStudy Class Reference	582
25.212.1	Detailed Description	583
25.212.2	Member Function Documentation	583
25.212.2.1	GetName	583
25.212.2.2	GetOptionalTags	583
25.212.2.3	GetQueryLevel	583
25.212.2.4	GetRequiredTags	583
25.212.2.5	GetUniqueTags	583
25.213.1	dcm::RAWCodec Class Reference	583
25.213.1	Detailed Description	585
25.213.2	Constructor & Destructor Documentation	585
25.213.2.1	RAWCodec	585
25.213.2.2	~RAWCodec	585
25.213.3	Member Function Documentation	585
25.213.3.1	CanCode	585
25.213.3.2	CanDecode	585
25.213.3.3	Code	585
25.213.3.4	Decode	585
25.213.3.5	Decode	585
25.213.3.6	DecodeBytes	586
25.213.3.7	GetHeaderInfo	586
25.214.1	dcm::Reader Class Reference	586
25.214.1	Detailed Description	588
25.214.2	Constructor & Destructor Documentation	589
25.214.2.1	Reader	589
25.214.2.2	~Reader	589
25.214.3	Member Function Documentation	589
25.214.3.1	CanRead	589
25.214.3.2	GetFile	589
25.214.3.3	GetFile	589
25.214.3.4	GetStreamPtr	589
25.214.3.5	Read	589
25.214.3.6	ReadDataSet	590
25.214.3.7	ReadMetaInformation	590
25.214.3.8	ReadPreamble	590

25.214.3.9	ReadSelectedTags	590
25.214.3.10	ReadUpToTag	590
25.214.3.11	SetFile	590
25.214.3.12	SetFileName	590
25.214.3.13	SetStream	590
25.214.4	Friends And Related Function Documentation	590
25.214.4.1	StreamImageReader	591
25.214.5	Member Data Documentation	591
25.214.5.1F	591
25.215	dcm::Rescaler Class Reference	591
25.215.1	Detailed Description	592
25.215.2	Constructor & Destructor Documentation	592
25.215.2.1	Rescaler	592
25.215.2.2	~Rescaler	592
25.215.3	Member Function Documentation	592
25.215.3.1	ComputeInterceptSlopePixelType	593
25.215.3.2	ComputePixelTypeFromMinMax	593
25.215.3.3	GetIntercept	593
25.215.3.4	GetSlope	593
25.215.3.5	InverseRescale	593
25.215.3.6	InverseRescaleFunctionIntoBestFit	593
25.215.3.7	Rescale	593
25.215.3.8	RescaleFunctionIntoBestFit	593
25.215.3.9	SetIntercept	593
25.215.3.10	SetMinMaxForPixelType	593
25.215.3.11	SetPixelFormat	593
25.215.3.12	SetSlope	593
25.215.3.13	SetTargetPixelType	593
25.215.3.14	SetUseTargetPixelType	594
25.216	dcm::RLECodec Class Reference	594
25.216.1	Detailed Description	595
25.216.2	Constructor & Destructor Documentation	595
25.216.2.1	RLECodec	595
25.216.2.2	~RLECodec	595
25.216.3	Member Function Documentation	595
25.216.3.1	CanCode	595
25.216.3.2	CanDecode	596

25.216.3.3	Code	596
25.216.3.4	Decode	596
25.216.3.5	Decode	596
25.216.3.6	GetBufferLength	596
25.216.3.7	GetHeaderInfo	596
25.216.3.8	SetBufferLength	596
25.216.3.9	SetLength	596
25.217	gdcmm::SerieHelper::Rule Struct Reference	596
25.217.1	Member Data Documentation	597
25.217.1.1	elem	597
25.217.1.2	group	597
25.217.1.3	op	597
25.217.1.4	value	597
25.218	gdcmm::Scanner Class Reference	597
25.218.1	Detailed Description	600
25.218.2	Member Typedef Documentation	600
25.218.2.1	ConstIterator	600
25.218.2.2	MappingType	600
25.218.2.3	TagToValue	600
25.218.2.4	TagToValueValueType	601
25.218.2.5	ValuesType	601
25.218.3	Constructor & Destructor Documentation	601
25.218.3.1	Scanner	601
25.218.3.2	~Scanner	601
25.218.4	Member Function Documentation	601
25.218.4.1	AddPrivateTag	601
25.218.4.2	AddSkipTag	601
25.218.4.3	AddTag	601
25.218.4.4	Begin	601
25.218.4.5	ClearSkipTags	601
25.218.4.6	ClearTags	601
25.218.4.7	End	601
25.218.4.8	GetAllFileNamesFromTagToValue	601
25.218.4.9	GetFilenameFromTagToValue	601
25.218.4.10	GetFileNames	601
25.218.4.11	GetKeys	601
25.218.4.12	GetMapping	602

25.218.4.10	GetMappingFromTagToValue	602
25.218.4.10	GetMappings	602
25.218.4.10	GetOrderedValues	602
25.218.4.10	GetValue	602
25.218.4.10	GetValues	602
25.218.4.10	GetValues	602
25.218.4.10	Key	603
25.218.4.20	New	603
25.218.4.20	Print	603
25.218.4.20	ProcessPublicTag	603
25.218.4.20	Scan	603
25.218.5	Friends And Related Function Documentation	603
25.218.5.1	operator<<	603
25.219	gdcm::Segment Class Reference	603
25.219.1	Detailed Description	605
25.219.2	Member Typedef Documentation	606
25.219.2.1	SurfaceVector	606
25.219.3	Member Enumeration Documentation	606
25.219.3.1	ALGOType	606
25.219.4	Constructor & Destructor Documentation	606
25.219.4.1	Segment	606
25.219.4.2	~Segment	606
25.219.5	Member Function Documentation	606
25.219.5.1	AddSurface	606
25.219.5.2	GetALGOType	606
25.219.5.3	GetALGOTypeString	606
25.219.5.4	GetAnatomicRegion	606
25.219.5.5	GetAnatomicRegion	606
25.219.5.6	GetPropertyCategory	606
25.219.5.7	GetPropertyCategory	606
25.219.5.8	GetPropertyType	606
25.219.5.9	GetPropertyType	606
25.219.5.10	GetSegmentAlgorithmName	606
25.219.5.10	GetSegmentAlgorithmType	606
25.219.5.10	GetSegmentDescription	606
25.219.5.10	GetSegmentLabel	606
25.219.5.10	GetSegmentNumber	606

25.219.5.16	GetSurface	606
25.219.5.16	GetSurfaceCount	607
25.219.5.16	GetSurfaces	607
25.219.5.16	GetSurfaces	607
25.219.5.19	SetAnatomicRegion	607
25.219.5.29	SetPropertyCategory	607
25.219.5.29	SetPropertyType	607
25.219.5.29	SetSegmentAlgorithmName	607
25.219.5.29	SetSegmentAlgorithmType	607
25.219.5.29	SetSegmentAlgorithmType	607
25.219.5.29	SetSegmentDescription	607
25.219.5.29	SetSegmentLabel	607
25.219.5.29	SetSegmentNumber	607
25.219.5.29	SetSurfaceCount	607
25.219.6	Member Data Documentation	607
25.219.6.1	AnatomicRegion	607
25.219.6.2	PropertyCategory	607
25.219.6.3	PropertyType	607
25.219.6.4	SegmentAlgorithmName	607
25.219.6.5	SegmentAlgorithmType	607
25.219.6.6	SegmentDescription	607
25.219.6.7	SegmentLabel	607
25.219.6.8	SegmentNumber	607
25.219.6.9	SurfaceCount	607
25.219.6.10	Surfaces	607
25.220	dcm::SegmentedPaletteColorLookupTable Class Reference	608
25.220.1	Detailed Description	609
25.220.2	Constructor & Destructor Documentation	609
25.220.2.1	SegmentedPaletteColorLookupTable	609
25.220.2.2	~SegmentedPaletteColorLookupTable	609
25.220.3	Member Function Documentation	609
25.220.3.1	Print	609
25.220.3.2	SetLUT	609
25.221	dcm::SegmentReader Class Reference	609
25.221.1	Detailed Description	611
25.221.2	Member Typedef Documentation	611
25.221.2.1	SegmentMap	611

25.221.2.2	SegmentVector	611
25.221.3	Constructor & Destructor Documentation	611
25.221.3.1	SegmentReader	611
25.221.3.2	~SegmentReader	611
25.221.4	Member Function Documentation	611
25.221.4.1	GetSegments	611
25.221.4.2	GetSegments	612
25.221.4.3	Read	612
25.221.4.4	ReadSegment	612
25.221.4.5	ReadSegments	612
25.221.5	Member Data Documentation	612
25.221.5.1	Segments	612
25.222	dcm::SegmentWriter Class Reference	612
25.222.1	Detailed Description	613
25.222.2	Member Typedef Documentation	614
25.222.2.1	SegmentVector	614
25.222.3	Constructor & Destructor Documentation	614
25.222.3.1	SegmentWriter	614
25.222.3.2	~SegmentWriter	614
25.222.4	Member Function Documentation	614
25.222.4.1	AddSegment	614
25.222.4.2	GetNumberOfSegments	614
25.222.4.3	GetSegment	614
25.222.4.4	GetSegments	614
25.222.4.5	GetSegments	614
25.222.4.6	PrepareWrite	614
25.222.4.7	SetNumberOfSegments	614
25.222.4.8	SetSegments	614
25.222.4.9	Write	614
25.222.5	Member Data Documentation	614
25.222.5.1	Segments	614
25.223	dcm::SequenceOfFragments Class Reference	614
25.223.1	Detailed Description	616
25.223.2	Member Typedef Documentation	617
25.223.2.1	ConstIterator	617
25.223.2.2	FragmentVector	617
25.223.2.3	Iterator	617

25.223.2.4	SizeType	617
25.223.3	Constructor & Destructor Documentation	617
25.223.3.1	SequenceOfFragments	617
25.223.4	Member Function Documentation	617
25.223.4.1	AddFragment	617
25.223.4.2	Begin	617
25.223.4.3	Begin	617
25.223.4.4	Clear	617
25.223.4.5	ComputeByteLength	617
25.223.4.6	ComputeLength	617
25.223.4.7	End	617
25.223.4.8	End	617
25.223.4.9	GetBuffer	617
25.223.4.10	GetFragBuffer	617
25.223.4.11	GetFragment	618
25.223.4.12	GetLength	618
25.223.4.13	GetNumberOfFragments	618
25.223.4.14	GetTable	618
25.223.4.15	GetTable	618
25.223.4.16	New	618
25.223.4.17	operator==	618
25.223.4.18	Print	618
25.223.4.19	Read	618
25.223.4.20	SetLength	618
25.223.4.21	Write	618
25.223.4.22	WriteBuffer	619
25.224	dcm::SequenceOfItems Class Reference	619
25.224.1	Detailed Description	621
25.224.2	Member Typedef Documentation	622
25.224.2.1	ConstIterator	622
25.224.2.2	ItemVector	622
25.224.2.3	Iterator	622
25.224.2.4	SizeType	622
25.224.3	Constructor & Destructor Documentation	622
25.224.3.1	SequenceOfItems	622
25.224.4	Member Function Documentation	622
25.224.4.1	AddItem	622

25.224.4.2	Begin	622
25.224.4.3	Begin	622
25.224.4.4	Clear	622
25.224.4.5	ComputeLength	622
25.224.4.6	End	622
25.224.4.7	End	622
25.224.4.8	FindDataElement	622
25.224.4.9	GetItem	622
25.224.4.10	GetItem	623
25.224.4.10	GetLength	623
25.224.4.10	GetNumberOfItems	623
25.224.4.11	UndefinedLength	623
25.224.4.11	New	623
25.224.4.15	operator=	623
25.224.4.16	operator==	623
25.224.4.17	Print	623
25.224.4.18	Read	623
25.224.4.19	SetLength	623
25.224.4.20	SetLengthToUndefined	624
25.224.4.23	SetNumberOfItems	624
25.224.4.24	Write	624
25.224.5	Member Data Documentation	624
25.224.5.1	Items	624
25.224.5.2	SequenceLengthField	624
25.225	dcm::SerieHelper Class Reference	624
25.225.1	Detailed Description	626
25.225.2	Member Typedef Documentation	626
25.225.2.1	SerieRestrictions	626
25.225.2.2	SingleSerieUIDFileSetmap	626
25.225.3	Constructor & Destructor Documentation	626
25.225.3.1	SerieHelper	626
25.225.3.2	~SerieHelper	626
25.225.4	Member Function Documentation	626
25.225.4.1	AddFile	626
25.225.4.2	AddFileName	626
25.225.4.3	AddRestriction	626
25.225.4.4	AddRestriction	626

25.225.4.5AddRestriction	627
25.225.4.6Clear	627
25.225.4.7CreateDefaultUniqueSeriesIdentifier	627
25.225.4.8CreateUniqueSeriesIdentifier	627
25.225.4.9FileNameOrdering	627
25.225.4.10GetFirstSingleSerieUIDFileSet	627
25.225.4.10GetNextSingleSerieUIDFileSet	627
25.225.4.11ImagePositionPatientOrdering	627
25.225.4.10OrderFileList	627
25.225.4.13SetDirectory	627
25.225.4.15SetLoadMode	627
25.225.4.16SetUseSeriesDetails	627
25.225.4.17UserOrdering	627
25.225.5Member Data Documentation	627
25.225.5.1ItFileSetHt	627
25.225.5.2SingleSerieUIDFileSetHT	627
25.226dcm::Series Class Reference	627
25.226.1Detailed Description	627
25.226.2Constructor & Destructor Documentation	628
25.226.2.1Series	628
25.227dcm::ServiceClassUser Class Reference	628
25.227.1Detailed Description	630
25.227.2Constructor & Destructor Documentation	630
25.227.2.1ServiceClassUser	630
25.227.2.2~ServiceClassUser	630
25.227.3Member Function Documentation	630
25.227.3.1GetAETitle	630
25.227.3.2GetCalledAETitle	630
25.227.3.3GetTimeout	630
25.227.3.4InitializeConnection	630
25.227.3.5SendEcho	631
25.227.3.6SendFind	631
25.227.3.7SendMove	631
25.227.3.8SendMove	631
25.227.3.9SendMove	631
25.227.3.10SendStore	631
25.227.3.10SendStore	631

25.227.3.18	SendStore	631
25.227.3.19	SetAETitle	631
25.227.3.19	SetCalledAETitle	631
25.227.3.19	SetHostname	632
25.227.3.19	SetPort	632
25.227.3.19	SetPortSCP	632
25.227.3.19	SetPresentationContexts	632
25.227.3.19	SetTimeout	632
25.227.3.20	StartAssociation	632
25.227.3.20	StopAssociation	633
25.228	dcm::SHA1 Class Reference	633
25.228.1	Detailed Description	633
25.228.2	Constructor & Destructor Documentation	633
25.228.2.1	SHA1	633
25.228.2.2	~SHA1	633
25.228.3	Member Function Documentation	633
25.228.3.1	Compute	634
25.228.3.2	ComputeFile	634
25.229	dcm::SimpleMemberCommand< T > Class Template Reference	634
25.229.1	Detailed Description	636
25.229.2	Member Typedef Documentation	636
25.229.2.1	Self	636
25.229.2.2	TMemberFunctionPointer	636
25.229.3	Constructor & Destructor Documentation	636
25.229.3.1	SimpleMemberCommand	636
25.229.3.2	~SimpleMemberCommand	636
25.229.4	Member Function Documentation	636
25.229.4.1	Execute	636
25.229.4.2	Execute	636
25.229.4.3	New	637
25.229.4.4	SetCallbackFunction	637
25.229.5	Member Data Documentation	637
25.229.5.1	m_MemberFunction	637
25.229.5.2	m_This	637
25.230	dcm::SimpleSubjectWatcher Class Reference	637
25.230.1	Detailed Description	638
25.230.2	Constructor & Destructor Documentation	638

25.230.2.1SimpleSubjectWatcher	638
25.230.2.2~SimpleSubjectWatcher	638
25.230.3Member Function Documentation	638
25.230.3.1EndFilter	638
25.230.3.2ShowAbort	638
25.230.3.3ShowAnonymization	638
25.230.3.4ShowData	638
25.230.3.5ShowDataSet	638
25.230.3.6ShowIteration	638
25.230.3.7ShowProgress	638
25.230.3.8StartFilter	638
25.230.3.9TestAbortOff	638
25.230.3.10TestAbortOn	638
25.230dcm::SmartPointer< ObjectType > Class Template Reference	639
25.231.1Detailed Description	640
25.231.2Constructor & Destructor Documentation	641
25.231.2.1SmartPointer	641
25.231.2.2SmartPointer	641
25.231.2.3SmartPointer	641
25.231.2.4SmartPointer	641
25.231.2.5~SmartPointer	641
25.231.3Member Function Documentation	641
25.231.3.1GetPointer	641
25.231.3.2operator ObjectType *	641
25.231.3.3operator*	641
25.231.3.4operator->	641
25.231.3.5operator=	641
25.231.3.6operator=	641
25.231.3.7operator=	642
25.230dcm::SOPClassUIDToIOD Class Reference	642
25.232.1Detailed Description	642
25.232.2Member Typedef Documentation	642
25.232.2.1const	642
25.232.3Member Function Documentation	642
25.232.3.1GetIOD	642
25.232.3.2GetIODFromSOPClassUID	643
25.232.3.3GetNumberOfSOPClassToIOD	643

25.232.3.4	GetSOPClassUIDFromIOD	643
25.232.3.5	GetSOPClassUIDToIOD	643
25.232.3.6	GetSOPClassUIDToIODs	643
25.233	dcm::Sorter Class Reference	643
25.233.1	Detailed Description	645
25.233.2	Member Typedef Documentation	645
25.233.2.1	SelectionMap	645
25.233.2.2	SortFunction	645
25.233.3	Constructor & Destructor Documentation	645
25.233.3.1	Sorter	645
25.233.3.2	~Sorter	645
25.233.4	Member Function Documentation	645
25.233.4.1	AddSelect	645
25.233.4.2	GetFileNames	646
25.233.4.3	Print	646
25.233.4.4	SetSortFunction	646
25.233.4.5	Sort	646
25.233.4.6	StableSort	646
25.233.5	Friends And Related Function Documentation	646
25.233.5.1	operator<<	646
25.233.6	Member Data Documentation	646
25.233.6.1	FileNames	646
25.233.6.2	Selection	646
25.233.6.3	SortFunc	647
25.234	dcm::Spacing Class Reference	647
25.234.1	Detailed Description	647
25.234.2	Member Enumeration Documentation	648
25.234.2.1	SpacingType	648
25.234.3	Constructor & Destructor Documentation	648
25.234.3.1	Spacing	648
25.234.3.2	~Spacing	648
25.234.4	Member Function Documentation	648
25.234.4.1	ComputePixelAspectRatioFromPixelSpacing	648
25.235	dcm::Spectroscopy Class Reference	649
25.235.1	Detailed Description	649
25.235.2	Constructor & Destructor Documentation	649
25.235.2.1	Spectroscopy	649

25.236	dcm::SplitMosaicFilter Class Reference	649
25.236.1	Detailed Description	649
25.236.2	Constructor & Destructor Documentation	650
25.236.2.1	SplitMosaicFilter	650
25.236.2.2	~SplitMosaicFilter	650
25.236.3	Member Function Documentation	650
25.236.3.1	ComputeMOSAICDimensions	650
25.236.3.2	GetFile	650
25.236.3.3	GetFile	650
25.236.3.4	GetImage	650
25.236.3.5	GetImage	650
25.236.3.6	SetFile	650
25.236.3.7	SetImage	650
25.236.3.8	Split	650
25.237	dcm::StartEvent Class Reference	650
25.238	dcm::static_assert_test< x > Struct Template Reference	651
25.239	dcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference	652
25.239.1	Member Enumeration Documentation	652
25.239.1.1	anonymous enum	652
25.240	dcm::StreamImageReader Class Reference	652
25.240.1	Detailed Description	654
25.240.2	Constructor & Destructor Documentation	654
25.240.2.1	StreamImageReader	654
25.240.2.2	~StreamImageReader	654
25.240.3	Member Function Documentation	654
25.240.3.1	CanReadImage	654
25.240.3.2	DefinePixelExtent	655
25.240.3.3	DefineProperBufferLength	655
25.240.3.4	GetDimensionsValueForResolution	655
25.240.3.5	GetFile	655
25.240.3.6	Read	655
25.240.3.7	ReadImageInformation	656
25.240.3.8	ReadImageSubregionJpegLS	656
25.240.3.9	ReadImageSubregionRAW	656
25.240.3.10	SetFileName	656
25.240.3.11	SetStream	656
25.240.4	Member Data Documentation	656

25.240.4.1mFileOffset	656
25.240.4.2mFileOffset1	656
25.240.4.3mHeaderInformation	656
25.240.4.4mReader	656
25.240.4.5mXMax	656
25.240.4.6mXMin	656
25.240.4.7mYMax	656
25.240.4.8mYMin	656
25.240.4.9mZMax	657
25.240.4.10mZMin	657
25.241.0dcm::StreamImageWriter Class Reference	657
25.241.1Detailed Description	659
25.241.2Constructor & Destructor Documentation	659
25.241.2.1StreamImageWriter	659
25.241.2.2~StreamImageWriter	659
25.241.3Member Function Documentation	659
25.241.3.1CanWriteFile	660
25.241.3.2DefinePixelExtent	660
25.241.3.3DefineProperBufferLength	660
25.241.3.4SetFile	660
25.241.3.5SetFileName	660
25.241.3.6SetStream	660
25.241.3.7Write	661
25.241.3.8WriteImageInformation	661
25.241.3.9WriteImageSubregionRAW	661
25.241.3.10WriteRawHeader	661
25.241.4Member Data Documentation	661
25.241.4.1mElementOffsets	661
25.241.4.2mElementOffsets1	662
25.241.4.3mspFile	662
25.241.4.4mWriter	662
25.241.4.5mXMax	662
25.241.4.6mXMin	662
25.241.4.7mYMax	662
25.241.4.8mYMin	662
25.241.4.9mZMax	662
25.241.4.10mZMin	662

25.242.0	dcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference	662
25.242.1	Detailed Description	664
25.242.2	Member Typedef Documentation	664
25.242.2.1	const_iterator	664
25.242.2.2	const_reference	664
25.242.2.3	const_reverse_iterator	664
25.242.2.4	difference_type	664
25.242.2.5	iterator	664
25.242.2.6	pointer	664
25.242.2.7	reference	664
25.242.2.8	reverse_iterator	664
25.242.2.9	size_type	664
25.242.2.10	value_type	664
25.242.3	Constructor & Destructor Documentation	664
25.242.3.1	String	665
25.242.3.2	String	665
25.242.3.3	String	665
25.242.3.4	String	665
25.242.4	Member Function Documentation	665
25.242.4.1	IsValid	665
25.242.4.2	operator const char *	665
25.242.4.3	Trim	665
25.242.4.4	Truncate	665
25.243.0	dcm::StringFilter Class Reference	665
25.243.1	Detailed Description	666
25.243.2	Constructor & Destructor Documentation	666
25.243.2.1	StringFilter	666
25.243.2.2	~StringFilter	666
25.243.3	Member Function Documentation	666
25.243.3.1	ExecuteQuery	666
25.243.3.2	ExecuteQuery	667
25.243.3.3	FromString	667
25.243.3.4	FromString	667
25.243.3.5	GetFile	667
25.243.3.6	GetFile	667
25.243.3.7	SetDicts	667
25.243.3.8	SetFile	667

25.243.3.9ToString	667
25.243.3.10ToStringPair	667
25.243.3.11ToStringPair	667
25.243.3.12UseDictAlways	667
25.244dcm::Study Class Reference	668
25.244.1Detailed Description	668
25.244.2Constructor & Destructor Documentation	668
25.244.2.1Study	668
25.245dcm::Subject Class Reference	668
25.245.1Detailed Description	669
25.245.2Constructor & Destructor Documentation	669
25.245.2.1Subject	669
25.245.2.2~Subject	669
25.245.3Member Function Documentation	669
25.245.3.1AddObserver	670
25.245.3.2AddObserver	670
25.245.3.3GetCommand	670
25.245.3.4HasObserver	670
25.245.3.5InvokeEvent	670
25.245.3.6InvokeEvent	670
25.245.3.7RemoveAllObservers	670
25.245.3.8RemoveObserver	670
25.246dcm::Surface Class Reference	670
25.246.1Detailed Description	673
25.246.2Member Enumeration Documentation	673
25.246.2.1STATES	673
25.246.2.2VIEWType	674
25.246.3Constructor & Destructor Documentation	674
25.246.3.1Surface	674
25.246.3.2~Surface	674
25.246.4Member Function Documentation	674
25.246.4.1GetAlgorithmFamily	674
25.246.4.2GetAlgorithmFamily	674
25.246.4.3GetAlgorithmName	674
25.246.4.4GetAlgorithmVersion	674
25.246.4.5GetAxisOfRotation	674
25.246.4.6GetCenterOfRotation	674

25.246.4.7	GetFiniteVolume	674
25.246.4.8	GetManifold	674
25.246.4.9	GetMaximumPointDistance	674
25.246.4.10	GetMeanPointDistance	674
25.246.4.10	GetMeshPrimitive	675
25.246.4.10	GetMeshPrimitive	675
25.246.4.10	GetNumberOfSurfacePoints	675
25.246.4.10	GetNumberOfVectors	675
25.246.4.10	GetPointCoordinatesData	675
25.246.4.10	GetPointCoordinatesData	675
25.246.4.10	GetPointPositionAccuracy	675
25.246.4.10	GetPointsBoundingBoxCoordinates	675
25.246.4.10	GetProcessingAlgorithm	675
25.246.4.20	GetProcessingAlgorithm	675
25.246.4.20	GetRecommendedDisplayCIELabValue	675
25.246.4.20	GetRecommendedDisplayCIELabValue	675
25.246.4.20	GetRecommendedDisplayGrayscaleValue	675
25.246.4.20	GetRecommendedPresentationOpacity	675
25.246.4.20	GetRecommendedPresentationType	675
25.246.4.20	GetSTATES	675
25.246.4.20	GetSTATESString	675
25.246.4.20	GetSurfaceComments	675
25.246.4.20	GetSurfaceNumber	675
25.246.4.30	GetSurfaceProcessing	675
25.246.4.30	GetSurfaceProcessingDescription	675
25.246.4.30	GetSurfaceProcessingRatio	675
25.246.4.30	GetVectorAccuracy	676
25.246.4.30	GetVectorCoordinateData	676
25.246.4.30	GetVectorCoordinateData	676
25.246.4.30	GetVectorDimensionality	676
25.246.4.30	GetVIEWType	676
25.246.4.30	GetVIEWTypeString	676
25.246.4.30	GetAlgorithmFamily	676
25.246.4.40	GetAlgorithmName	676
25.246.4.40	GetAlgorithmVersion	676
25.246.4.40	GetAxisOfRotation	676
25.246.4.40	GetCenterOfRotation	676

25.246.4.49	SetFiniteVolume	676
25.246.4.49	SetManifold	676
25.246.4.49	SetMaximumPointDistance	676
25.246.4.49	SetMeanPointDistance	676
25.246.4.49	SetMeshPrimitive	676
25.246.4.49	SetNumberOfSurfacePoints	676
25.246.4.50	SetNumberOfVectors	676
25.246.4.50	SetPointCoordinatesData	676
25.246.4.50	SetPointPositionAccuracy	676
25.246.4.50	SetPointsBoundingBoxCoordinates	676
25.246.4.50	SetProcessingAlgorithm	676
25.246.4.55	SetRecommendedDisplayCIELabValue	676
25.246.4.56	SetRecommendedDisplayCIELabValue	676
25.246.4.57	SetRecommendedDisplayCIELabValue	676
25.246.4.58	SetRecommendedDisplayGrayscaleValue	676
25.246.4.59	SetRecommendedPresentationOpacity	677
25.246.4.60	SetRecommendedPresentationType	677
25.246.4.63	SetSurfaceComments	677
25.246.4.62	SetSurfaceNumber	677
25.246.4.63	SetSurfaceProcessing	677
25.246.4.63	SetSurfaceProcessingDescription	677
25.246.4.65	SetSurfaceProcessingRatio	677
25.246.4.66	SetVectorAccuracy	677
25.246.4.67	SetVectorCoordinateData	677
25.246.4.68	SetVectorDimensionality	677
25.247	dcm::SurfaceHelper Class Reference	677
25.247.1	Member Typedef Documentation	678
25.247.1.1	ColorArray	678
25.247.2	Member Function Documentation	678
25.247.2.1	RecommendedDisplayCIELabToRGB	678
25.247.2.2	RecommendedDisplayCIELabToRGB	678
25.247.2.3	RGBToRecommendedDisplayCIELab	678
25.247.2.4	RGBToRecommendedDisplayGrayscale	679
25.248	dcm::SurfaceReader Class Reference	679
25.248.1	Detailed Description	681
25.248.2	Constructor & Destructor Documentation	681
25.248.2.1	SurfaceReader	681

25.248.2.2~SurfaceReader	681
25.248.3Member Function Documentation	681
25.248.3.1GetNumberOfSurfaces	681
25.248.3.2Read	681
25.248.3.3ReadPointMacro	681
25.248.3.4ReadSurface	681
25.248.3.5ReadSurfaces	681
25.249dcm::SurfaceWriter Class Reference	682
25.249.1Detailed Description	683
25.249.2Constructor & Destructor Documentation	683
25.249.2.1SurfaceWriter	683
25.249.2.2~SurfaceWriter	683
25.249.3Member Function Documentation	683
25.249.3.1ComputeNumberOfSurfaces	683
25.249.3.2GetNumberOfSurfaces	683
25.249.3.3PrepareWrite	683
25.249.3.4PrepareWritePointMacro	683
25.249.3.5SetNumberOfSurfaces	683
25.249.3.6Write	683
25.249.4Member Data Documentation	683
25.249.4.1NumberOfSurfaces	683
25.250dcm::SwapCode Class Reference	684
25.250.1Detailed Description	684
25.250.2Member Enumeration Documentation	684
25.250.2.1SwapCodeType	684
25.250.3Constructor & Destructor Documentation	685
25.250.3.1SwapCode	685
25.250.4Member Function Documentation	685
25.250.4.1GetIndex	685
25.250.4.2GetSwapCodeString	685
25.250.4.3operator SwapCode::SwapCodeType	685
25.250.5Friends And Related Function Documentation	685
25.250.5.1operator<<	685
25.251dcm::SwapperDoOp Class Reference	685
25.251.1Member Function Documentation	685
25.251.1.1Swap	685
25.251.1.2SwapArray	685

25.252dcm::SwapperNoOp Class Reference	686
25.252.1Detailed Description	686
25.252.2Member Function Documentation	686
25.252.2.1Swap	686
25.252.2.2SwapArray	686
25.253dcm::System Class Reference	686
25.253.1Detailed Description	687
25.253.2Member Function Documentation	687
25.253.2.1DeleteDirectory	687
25.253.2.2EncodeBytes	687
25.253.2.3FileExists	688
25.253.2.4FileIsDirectory	688
25.253.2.5FileIsSymlink	688
25.253.2.6FileSize	688
25.253.2.7FileTime	688
25.253.2.8FormatDateTime	688
25.253.2.9GetCurrentDateTime	688
25.253.2.10GetCurrentModuleFileName	689
25.253.2.11GetCurrentProcessFileName	689
25.253.2.12GetCurrentResourcesDirectory	689
25.253.2.13GetCWD	689
25.253.2.14GetHostName	689
25.253.2.15GetLastError	689
25.253.2.16GetLocaleCharset	689
25.253.2.17GetPermissions	689
25.253.2.18GetTimezoneOffsetFromUTC	689
25.253.2.19MakeDirectory	689
25.253.2.20ParseDateTime	690
25.253.2.21ParseDateTime	690
25.253.2.22RemoveFile	690
25.253.2.23SetPermissions	690
25.253.2.24StrCaseCmp	690
25.253.2.25StrNCaseCmp	690
25.253.2.26StrTokR	690
25.254dcm::Table Class Reference	690
25.254.1Detailed Description	691
25.254.2Member Typedef Documentation	691

25.254.2.1	MapTableEntry	691
25.254.3	Constructor & Destructor Documentation	691
25.254.3.1	Table	691
25.254.3.2	~Table	691
25.254.4	Member Function Documentation	691
25.254.4.1	GetTableEntry	691
25.254.4.2	InsertEntry	691
25.254.5	Friends And Related Function Documentation	691
25.254.5.1	operator<<	691
25.255	dcm::TableEntry Class Reference	691
25.255.1	Detailed Description	692
25.255.2	Constructor & Destructor Documentation	692
25.255.2.1	TableEntry	692
25.255.2.2	~TableEntry	692
25.256	dcm::TableReader Class Reference	692
25.256.1	Detailed Description	693
25.256.2	Constructor & Destructor Documentation	693
25.256.2.1	TableReader	693
25.256.2.2	~TableReader	693
25.256.3	Member Function Documentation	693
25.256.3.1	CharacterDataHandler	693
25.256.3.2	EndElement	693
25.256.3.3	GetDefs	693
25.256.3.4	GetFilename	693
25.256.3.5	HandleIOD	693
25.256.3.6	HandleIODEntry	693
25.256.3.7	HandleMacro	693
25.256.3.8	HandleMacroEntry	693
25.256.3.9	HandleMacroEntryDescription	693
25.256.3.10	HandleModule	693
25.256.3.11	HandleModuleEntry	693
25.256.3.12	HandleModuleEntryDescription	694
25.256.3.13	HandleModuleInclude	694
25.256.3.14	Read	694
25.256.3.15	SetFilename	694
25.256.3.16	StartElement	694
25.257	dcm::network::TableRow Class Reference	694

25.257.1	Member Data Documentation	694
25.257.1.1	transitions	695
25.258	dcm::Tag Class Reference	695
25.258.1	Detailed Description	696
25.258.2	Constructor & Destructor Documentation	697
25.258.2.1	Tag	697
25.258.2.2	Tag	697
25.258.2.3	Tag	697
25.258.3	Member Function Documentation	697
25.258.3.1	GetElement	697
25.258.3.2	GetElementTag	697
25.258.3.3	GetGroup	698
25.258.3.4	GetLength	698
25.258.3.5	GetPrivateCreator	698
25.258.3.6	IsGroupLength	698
25.258.3.7	IsGroupXX	698
25.258.3.8	IsIllegal	698
25.258.3.9	IsPrivate	698
25.258.3.10	IsPrivateCreator	699
25.258.3.11	IsPublic	699
25.258.3.12	operator!=	699
25.258.3.13	operator<	699
25.258.3.14	operator<=	699
25.258.3.15	operator=	699
25.258.3.16	operator==	699
25.258.3.17	operator[]	699
25.258.3.18	operator[]	699
25.258.3.19	PrintAsPipeSeparatedString	699
25.258.3.20	Read	700
25.258.3.21	ReadFromCommaSeparatedString	700
25.258.3.22	ReadFromPipeSeparatedString	700
25.258.3.23	SetElement	700
25.258.3.24	SetElementTag	700
25.258.3.25	SetElementTag	700
25.258.3.26	SetGroup	700
25.258.3.27	SetPrivateCreator	701
25.258.3.28	Write	701

25.258.4	Friends And Related Function Documentation	701
25.258.4.1	operator<<	701
25.258.4.2	operator>>	701
25.258.5	Member Data Documentation	701
25.258.5.1	bytes	701
25.258.5.2	tag	701
25.258.5.3	tags	701
25.259	gdcmm::TagPath Class Reference	701
25.259.1	Detailed Description	702
25.259.2	Constructor & Destructor Documentation	702
25.259.2.1	TagPath	702
25.259.2.2	~TagPath	702
25.259.3	Member Function Documentation	702
25.259.3.1	ConstructFromString	702
25.259.3.2	ConstructFromTagList	702
25.259.3.3	IsValid	702
25.259.3.4	Print	702
25.259.3.5	Push	702
25.259.3.6	Push	702
25.260	gdcmm::Testing Class Reference	703
25.260.1	Detailed Description	704
25.260.2	Member Typedef Documentation	704
25.260.2.1	MD5DataImagesType	704
25.260.2.2	MediaStorageDataFilesType	704
25.260.3	Constructor & Destructor Documentation	704
25.260.3.1	Testing	704
25.260.3.2	~Testing	704
25.260.4	Member Function Documentation	704
25.260.4.1	ComputeFileMD5	704
25.260.4.2	ComputeMD5	704
25.260.4.3	GetDataExtraRoot	704
25.260.4.4	GetDataRoot	705
25.260.4.5	GetFileName	705
25.260.4.6	GetFileNames	705
25.260.4.7	GetLossyFlagFromFile	705
25.260.4.8	GetMD5DataImage	705
25.260.4.9	GetMD5DataImages	705

25.260.4.10	GetMD5FromBrokenFile	705
25.260.4.10	GetMD5FromFile	705
25.260.4.10	GetMediaStorageDataFile	705
25.260.4.10	GetMediaStorageDataFiles	705
25.260.4.10	GetMediaStorageFromFile	705
25.260.4.10	GetNumberOfFileNames	705
25.260.4.10	GetNumberOfMD5DataImages	705
25.260.4.10	GetNumberOfMediaStorageDataFiles	706
25.260.4.10	GetPixelSpacingDataRoot	706
25.260.4.10	GetSelectedTagsOffsetFromFile	706
25.260.4.20	GetSourceDirectory	706
25.260.4.20	GetStreamOffsetFromFile	706
25.260.4.20	GetTempDirectory	706
25.260.4.20	GetTempDirectoryW	706
25.260.4.20	GetTempFilename	706
25.260.4.20	GetTempFilenameW	706
25.260.4.20	Print	706
25.261	dcm::Trace Class Reference	706
25.261.1	Detailed Description	707
25.261.2	Constructor & Destructor Documentation	707
25.261.2.1	Trace	707
25.261.2.2	~Trace	707
25.261.3	Member Function Documentation	707
25.261.3.1	DebugOff	707
25.261.3.2	DebugOn	707
25.261.3.3	ErrorOff	708
25.261.3.4	ErrorOn	708
25.261.3.5	GetDebugFlag	708
25.261.3.6	GetErrorFlag	708
25.261.3.7	GetStream	708
25.261.3.8	GetWarningFlag	708
25.261.3.9	SetDebug	708
25.261.3.10	SetError	708
25.261.3.11	SetStream	708
25.261.3.12	SetWarning	708
25.261.3.13	WarningOff	708
25.261.3.14	WarningOn	708

25.262	dcm::TransferSyntax Class Reference	708
25.262.1	Detailed Description	710
25.262.2	Member Enumeration Documentation	710
25.262.2.1	NegotiatedType	710
25.262.2.2	TSType	710
25.262.3	Constructor & Destructor Documentation	711
25.262.3.1	TransferSyntax	711
25.262.4	Member Function Documentation	711
25.262.4.1	CanStoreLossy	711
25.262.4.2	GetNegotiatedType	711
25.262.4.3	GetString	711
25.262.4.4	GetSwapCode	711
25.262.4.5	GetTSString	711
25.262.4.6	GetTSType	712
25.262.4.7	IsEncapsulated	712
25.262.4.8	IsEncoded	712
25.262.4.9	IsExplicit	712
25.262.4.10	IsImplicit	712
25.262.4.11	IsLossless	712
25.262.4.12	IsLossy	712
25.262.4.13	IsValid	712
25.262.4.14	operator TSType	712
25.262.5	Friends And Related Function Documentation	712
25.262.5.1	operator<<	712
25.263	dcm::network::TransferSyntaxSub Class Reference	712
25.263.1	Detailed Description	713
25.263.2	Constructor & Destructor Documentation	713
25.263.2.1	TransferSyntaxSub	713
25.263.3	Member Function Documentation	713
25.263.3.1	GetName	713
25.263.3.2	operator==	713
25.263.3.3	Print	713
25.263.3.4	Read	713
25.263.3.5	SetName	713
25.263.3.6	SetNameFromUID	713
25.263.3.7	Size	713
25.263.3.8	Write	713

25.264	dcm::network::Transition Struct Reference	713
25.264.1	Constructor & Destructor Documentation	714
25.264.1.1	Transition	714
25.264.1.2	~Transition	714
25.264.1.3	Transition	714
25.264.2	Member Function Documentation	715
25.264.2.1	MakeNew	715
25.264.3	Member Data Documentation	715
25.264.3.1	mAction	715
25.264.3.2	mEnd	715
25.265	dcm::Type Class Reference	715
25.265.1	Detailed Description	716
25.265.2	Member Enumeration Documentation	716
25.265.2.1	TypeType	716
25.265.3	Constructor & Destructor Documentation	716
25.265.3.1	Type	716
25.265.4	Member Function Documentation	716
25.265.4.1	GetTypeString	716
25.265.4.2	GetTypeType	716
25.265.4.3	operator TypeType	717
25.265.5	Friends And Related Function Documentation	717
25.265.5.1	operator<<	717
25.266	dcm::UI Struct Reference	717
25.266.1	Friends And Related Function Documentation	717
25.266.1.1	operator<<	717
25.266.2	Member Data Documentation	717
25.266.2.1	Internal	717
25.267	dcm::UIDGenerator Class Reference	717
25.267.1	Detailed Description	718
25.267.2	Constructor & Destructor Documentation	718
25.267.2.1	UIDGenerator	718
25.267.3	Member Function Documentation	718
25.267.3.1	Generate	718
25.267.3.2	GenerateUUID	718
25.267.3.3	GetGDCMUID	718
25.267.3.4	GetRoot	719
25.267.3.5	sValid	719

25.267.3.6SetRoot	719
25.268dcm::UIDs Class Reference	719
25.268.1Detailed Description	724
25.268.2Member Typedef Documentation	724
25.268.2.1TransferSyntaxStringsType	724
25.268.3Member Enumeration Documentation	724
25.268.3.1TSName	724
25.268.3.2TSType	731
25.268.4Member Function Documentation	737
25.268.4.1GetName	737
25.268.4.2GetNumberOfTransferSyntaxStrings	738
25.268.4.3GetString	738
25.268.4.4GetTransferSyntaxString	738
25.268.4.5GetTransferSyntaxStrings	738
25.268.4.6GetUIDName	738
25.268.4.7GetUIDString	738
25.268.4.8operator TSType	738
25.268.4.9SetFromUID	738
25.269dcm::network::ULAction Class Reference	738
25.269.1Detailed Description	740
25.269.2Constructor & Destructor Documentation	740
25.269.2.1ULAction	740
25.269.2.2~ULAction	740
25.269.3Member Function Documentation	740
25.269.3.1PerformAction	740
25.270dcm::network::ULActionAA1 Class Reference	741
25.270.1Member Function Documentation	741
25.270.1.1PerformAction	741
25.271dcm::network::ULActionAA2 Class Reference	742
25.271.1Member Function Documentation	742
25.271.1.1PerformAction	743
25.272dcm::network::ULActionAA3 Class Reference	743
25.272.1Member Function Documentation	744
25.272.1.1PerformAction	744
25.273dcm::network::ULActionAA4 Class Reference	744
25.273.1Member Function Documentation	745
25.273.1.1PerformAction	745

25.274	dcm::network::ULActionAA5 Class Reference	745
25.274.1	Member Function Documentation	746
25.274.1.1	PerformAction	746
25.275	dcm::network::ULActionAA6 Class Reference	746
25.275.1	Member Function Documentation	747
25.275.1.1	PerformAction	747
25.276	dcm::network::ULActionAA7 Class Reference	748
25.276.1	Member Function Documentation	748
25.276.1.1	PerformAction	748
25.277	dcm::network::ULActionAA8 Class Reference	749
25.277.1	Member Function Documentation	749
25.277.1.1	PerformAction	750
25.278	dcm::network::ULActionAE1 Class Reference	750
25.278.1	Member Function Documentation	751
25.278.1.1	PerformAction	751
25.279	dcm::network::ULActionAE2 Class Reference	751
25.279.1	Member Function Documentation	752
25.279.1.1	PerformAction	752
25.280	dcm::network::ULActionAE3 Class Reference	752
25.280.1	Member Function Documentation	753
25.280.1.1	PerformAction	753
25.281	dcm::network::ULActionAE4 Class Reference	753
25.281.1	Member Function Documentation	754
25.281.1.1	PerformAction	754
25.282	dcm::network::ULActionAE5 Class Reference	755
25.282.1	Member Function Documentation	755
25.282.1.1	PerformAction	755
25.283	dcm::network::ULActionAE6 Class Reference	756
25.283.1	Member Function Documentation	756
25.283.1.1	PerformAction	757
25.284	dcm::network::ULActionAE7 Class Reference	757
25.284.1	Member Function Documentation	758
25.284.1.1	PerformAction	758
25.285	dcm::network::ULActionAE8 Class Reference	758
25.285.1	Member Function Documentation	759
25.285.1.1	PerformAction	759
25.286	dcm::network::ULActionAR1 Class Reference	759

25.286.1Member Function Documentation	760
25.286.1.1PerformAction	760
25.287gdcmm::network::ULActionAR10 Class Reference	760
25.287.1Member Function Documentation	761
25.287.1.1PerformAction	761
25.288gdcmm::network::ULActionAR2 Class Reference	762
25.288.1Member Function Documentation	762
25.288.1.1PerformAction	762
25.289gdcmm::network::ULActionAR3 Class Reference	763
25.289.1Member Function Documentation	763
25.289.1.1PerformAction	764
25.290gdcmm::network::ULActionAR4 Class Reference	764
25.290.1Member Function Documentation	765
25.290.1.1PerformAction	765
25.291gdcmm::network::ULActionAR5 Class Reference	765
25.291.1Member Function Documentation	766
25.291.1.1PerformAction	766
25.292gdcmm::network::ULActionAR6 Class Reference	766
25.292.1Member Function Documentation	767
25.292.1.1PerformAction	767
25.293gdcmm::network::ULActionAR7 Class Reference	767
25.293.1Member Function Documentation	768
25.293.1.1PerformAction	768
25.294gdcmm::network::ULActionAR8 Class Reference	769
25.294.1Member Function Documentation	769
25.294.1.1PerformAction	769
25.295gdcmm::network::ULActionAR9 Class Reference	770
25.295.1Member Function Documentation	770
25.295.1.1PerformAction	771
25.296gdcmm::network::ULActionDT1 Class Reference	771
25.296.1Member Function Documentation	772
25.296.1.1PerformAction	772
25.297gdcmm::network::ULActionDT2 Class Reference	772
25.297.1Member Function Documentation	773
25.297.1.1PerformAction	773
25.298gdcmm::network::ULBasicCallback Class Reference	773
25.298.1Detailed Description	774

25.298.2	Constructor & Destructor Documentation	774
25.298.2.1	ULBasicCallback	774
25.298.2.2	~ULBasicCallback	774
25.298.3	Member Function Documentation	774
25.298.3.1	GetDataSets	774
25.298.3.2	HandleDataSet	774
25.299	dcm::network::ULConnection Class Reference	775
25.299.1	Detailed Description	775
25.299.2	Constructor & Destructor Documentation	776
25.299.2.1	ULConnection	776
25.299.2.2	~ULConnection	776
25.299.3	Member Function Documentation	776
25.299.3.1	AddAcceptedPresentationContext	776
25.299.3.2	FindContext	776
25.299.3.3	GetAcceptedPresentationContexts	776
25.299.3.4	GetAcceptedPresentationContexts	776
25.299.3.5	GetConnectionInfo	776
25.299.3.6	GetMaxPDUSize	776
25.299.3.7	GetPresentationContextACByID	776
25.299.3.8	GetPresentationContextIDFromPresentationContext	776
25.299.3.9	GetPresentationContextRQByID	776
25.299.3.10	GetPresentationContexts	776
25.299.3.11	GetProtocol	776
25.299.3.12	GetState	776
25.299.3.13	GetTimer	776
25.299.3.14	InitializeConnection	776
25.299.3.15	InitializeIncomingConnection	777
25.299.3.16	SetMaxPDUSize	777
25.299.3.17	SetPresentationContexts	777
25.299.3.18	SetPresentationContexts	777
25.299.3.19	SetState	777
25.299.3.20	StopProtocol	777
25.300	dcm::network::ULConnectionCallback Class Reference	777
25.300.1	Detailed Description	778
25.300.2	Constructor & Destructor Documentation	778
25.300.2.1	ULConnectionCallback	778
25.300.2.2	~ULConnectionCallback	778

25.300.3	Member Function Documentation	778
25.300.3.1	DataSetHandled	778
25.300.3.2	DataSetHandles	778
25.300.3.3	HandleDataSet	778
25.300.3.4	ResetHandledDataSet	778
25.301	dcm::network::ULConnectionInfo Class Reference	778
25.301.1	Detailed Description	779
25.301.2	Constructor & Destructor Documentation	779
25.301.2.1	ULConnectionInfo	779
25.301.3	Member Function Documentation	779
25.301.3.1	GetCalledAETitle	779
25.301.3.2	GetCalledComputerName	779
25.301.3.3	GetCalledIPAddress	779
25.301.3.4	GetCalledIPPort	779
25.301.3.5	GetCallingAETitle	779
25.301.3.6	GetMaxPDULength	779
25.301.3.7	GetUserInfo	779
25.301.3.8	Initialize	779
25.301.3.9	SetMaxPDULength	779
25.302	dcm::network::ULConnectionManager Class Reference	779
25.302.1	Detailed Description	781
25.302.2	Constructor & Destructor Documentation	781
25.302.2.1	ULConnectionManager	781
25.302.2.2	~ULConnectionManager	781
25.302.3	Member Function Documentation	781
25.302.3.1	BreakConnection	781
25.302.3.2	BreakConnectionNow	781
25.302.3.3	EstablishConnection	781
25.302.3.4	EstablishConnectionMove	782
25.302.3.5	SendEcho	782
25.302.3.6	SendFind	782
25.302.3.7	SendFind	782
25.302.3.8	SendMove	782
25.302.3.9	SendMove	782
25.302.3.10	SendStore	782
25.302.3.11	SendStore	782
25.303	dcm::network::ULEvent Class Reference	782

25.303.1	Detailed Description	782
25.303.2	Constructor & Destructor Documentation	783
25.303.2.1	ULEvent	783
25.303.2.2	ULEvent	783
25.303.2.3	~ULEvent	783
25.303.3	Member Function Documentation	783
25.303.3.1	GetEvent	783
25.303.3.2	GetPDUs	783
25.303.3.3	SetEvent	783
25.303.3.4	SetPDU	783
25.304	dcm::network::ULTransitionTable Class Reference	783
25.304.1	Detailed Description	783
25.304.2	Constructor & Destructor Documentation	783
25.304.2.1	ULTransitionTable	784
25.304.3	Member Function Documentation	784
25.304.3.1	HandleEvent	784
25.304.3.2	PrintTable	784
25.305	dcm::network::ULWritingCallback Class Reference	784
25.305.1	Constructor & Destructor Documentation	785
25.305.1.1	ULWritingCallback	785
25.305.1.2	~ULWritingCallback	785
25.305.2	Member Function Documentation	785
25.305.2.1	HandleDataSet	785
25.305.2.2	SetDirectory	785
25.306	dcm::UNExplicitDataElement Class Reference	786
25.306.1	Detailed Description	787
25.306.2	Member Function Documentation	787
25.306.2.1	GetLength	787
25.306.2.2	Read	787
25.306.2.3	ReadPreValue	787
25.306.2.4	ReadValue	787
25.306.2.5	ReadWithLength	787
25.307	dcm::UNExplicitImplicitDataElement Class Reference	787
25.307.1	Detailed Description	789
25.307.2	Member Function Documentation	789
25.307.2.1	GetLength	789
25.307.2.2	Read	789

25.307.2.3	ReadPreValue	789
25.307.2.4	ReadValue	789
25.308	dcm::Unpacker12Bits Class Reference	789
25.308.1	Detailed Description	789
25.308.2	Member Function Documentation	790
25.308.2.1	Pack	790
25.308.2.2	Unpack	790
25.309	dcm::Usage Class Reference	790
25.309.1	Detailed Description	791
25.309.2	Member Enumeration Documentation	791
25.309.2.1	UsageType	791
25.309.3	Constructor & Destructor Documentation	791
25.309.3.1	Usage	791
25.309.4	Member Function Documentation	791
25.309.4.1	GetUsageString	791
25.309.4.2	GetUsageType	792
25.309.4.3	operator UsageType	792
25.309.5	Friends And Related Function Documentation	792
25.309.5.1	operator<<	792
25.310	dcm::UserEvent Class Reference	792
25.311	dcm::network::UserInformation Class Reference	793
25.311.1	Detailed Description	793
25.311.2	Constructor & Destructor Documentation	794
25.311.2.1	UserInformation	794
25.311.3	Member Function Documentation	794
25.311.3.1	GetMaximumLengthSub	794
25.311.3.2	GetMaximumLengthSub	794
25.311.3.3	Print	794
25.311.3.4	Read	794
25.311.3.5	Size	794
25.311.3.6	Write	794
25.312	dcm::Validate Class Reference	794
25.312.1	Detailed Description	795
25.312.2	Constructor & Destructor Documentation	795
25.312.2.1	Validate	795
25.312.2.2	~Validate	795
25.312.3	Member Function Documentation	795

25.312.3.1	GetValidatedFile	795
25.312.3.2	SetFile	795
25.312.3.3	Validation	795
25.312.4	Member Data Documentation	795
25.312.4.1	F	795
25.312.4.2	V	795
25.313	dcm::Value Class Reference	795
25.313.1	Detailed Description	796
25.313.2	Constructor & Destructor Documentation	797
25.313.2.1	Value	797
25.313.2.2	~Value	797
25.313.3	Member Function Documentation	797
25.313.3.1	Clear	797
25.313.3.2	GetLength	797
25.313.3.3	operator==	797
25.313.3.4	SetLength	797
25.314	dcm::ValueIO< TDE, TSwap, TType > Class Template Reference	797
25.314.1	Detailed Description	797
25.314.2	Member Function Documentation	798
25.314.2.1	Read	798
25.314.2.2	Write	798
25.315	dcm::Version Class Reference	798
25.315.1	Detailed Description	798
25.315.2	Constructor & Destructor Documentation	798
25.315.2.1	Version	798
25.315.2.2	~Version	798
25.315.3	Member Function Documentation	798
25.315.3.1	GetBuildVersion	799
25.315.3.2	GetMajorVersion	799
25.315.3.3	GetMinorVersion	799
25.315.3.4	GetVersion	799
25.315.3.5	Print	799
25.315.4	Friends And Related Function Documentation	799
25.315.4.1	operator<<	799
25.316	dcm::VL Class Reference	799
25.316.1	Detailed Description	800
25.316.2	Member Typedef Documentation	800

25.316.2.1Type	800
25.316.3Constructor & Destructor Documentation	800
25.316.3.1VL	800
25.316.4Member Function Documentation	800
25.316.4.1GetLength	800
25.316.4.2GetVL16Max	800
25.316.4.3GetVL32Max	800
25.316.4.4IsOdd	800
25.316.4.5IsUndefined	800
25.316.4.6operator uint32_t	801
25.316.4.7operator++	801
25.316.4.8operator++	801
25.316.4.9operator+=	801
25.316.4.10Read	801
25.316.4.11Read16	801
25.316.4.12SetToUndefined	801
25.316.4.13Write	801
25.316.4.14Write16	801
25.316.5Friends And Related Function Documentation	801
25.316.5.1operator<<	801
25.317dcm::VM Class Reference	801
25.317.1Detailed Description	803
25.317.2Member Enumeration Documentation	803
25.317.2.1VMType	803
25.317.3Constructor & Destructor Documentation	804
25.317.3.1VM	804
25.317.4Member Function Documentation	804
25.317.4.1Compatible	804
25.317.4.2GetIndex	804
25.317.4.3GetLength	804
25.317.4.4GetNumberOfElementsFromArray	804
25.317.4.5GetVMString	804
25.317.4.6GetVMType	805
25.317.4.7GetVMTypeFromLength	805
25.317.4.8IsValid	805
25.317.4.9operator VMType	805
25.317.5Friends And Related Function Documentation	805

25.317.5.operator<<	805
25.318.dcm::VR Class Reference	805
25.318.1.Detailed Description	807
25.318.2.Member Enumeration Documentation	807
25.318.2.1.VRType	807
25.318.3.Constructor & Destructor Documentation	808
25.318.3.1.VR	808
25.318.4.Member Function Documentation	808
25.318.4.1.CanDisplay	808
25.318.4.2.Compatible	808
25.318.4.3.GetLength	809
25.318.4.4.GetLength	809
25.318.4.5.GetSize	809
25.318.4.6.GetSizeof	809
25.318.4.7.GetVRString	809
25.318.4.8.GetVRStringFromFile	809
25.318.4.9.GetVRType	809
25.318.4.10.GetVRTypeFromFile	809
25.318.4.11.ASCII	809
25.318.4.12.ASCII2	809
25.318.4.13.Binary	809
25.318.4.14.Binary2	809
25.318.4.15.Dual	809
25.318.4.16.Swap	809
25.318.4.17.Valid	809
25.318.4.18.Valid	809
25.318.4.19.VRFile	809
25.318.4.20.operator VRType	809
25.318.4.21.Read	809
25.318.4.22.Write	809
25.318.5.Friends And Related Function Documentation	810
25.318.5.operator<<	810
25.319.dcm::VR16ExplicitDataElement Class Reference	810
25.319.1.Detailed Description	811
25.319.2.Member Function Documentation	811
25.319.2.1.GetLength	811
25.319.2.2.Read	812

25.319.2.3	ReadPreValue	812
25.319.2.4	ReadValue	812
25.319.2.5	ReadWithLength	812
25.320	dcm::VRVLSize< 0 > Class Template Reference	812
25.320.1	Member Function Documentation	812
25.320.1.1	Read	812
25.320.1.2	Write	812
25.321	dcm::VRVLSize< 1 > Class Template Reference	812
25.321.1	Member Function Documentation	813
25.321.1.1	Read	813
25.321.1.2	Write	813
25.322	vtkGDCMImageReader Class Reference	813
25.322.1	Detailed Description	815
25.322.2	Constructor & Destructor Documentation	815
25.322.2.1	vtkGDCMImageReader	815
25.322.2.2	~vtkGDCMImageReader	815
25.322.3	Member Function Documentation	815
25.322.3.1	CanReadFile	815
25.322.3.2	ExecuteData	815
25.322.3.3	ExecuteInformation	815
25.322.3.4	FillMedicalImageInformation	816
25.322.3.5	GetDescriptiveName	816
25.322.3.6	GetFileExtensions	816
25.322.3.7	GetIconImage	816
25.322.3.8	GetOverlay	816
25.322.3.9	LoadSingleFile	816
25.322.3.10	New	816
25.322.3.11	PrintSelf	816
25.322.3.12	RequestDataCompat	816
25.322.3.13	RequestInformationCompat	816
25.322.3.14	SetCurve	816
25.322.3.15	SetFileNames	816
25.322.3.16	SetFilePattern	816
25.322.3.17	SetFilePrefix	816
25.322.3.18	SetMedicalImageProperties	816
25.322.3.19	SetBooleanMacro	817
25.322.3.20	SetBooleanMacro	817

25.322.3.21tkBooleanMacro	817
25.322.3.22tkBooleanMacro	817
25.322.3.23tkBooleanMacro	817
25.322.3.24tkGetMacro	817
25.322.3.25tkGetMacro	817
25.322.3.26tkGetMacro	817
25.322.3.27tkGetMacro	817
25.322.3.28tkGetMacro	817
25.322.3.29tkGetMacro	817
25.322.3.30tkGetMacro	817
25.322.3.31tkGetMacro	817
25.322.3.32tkGetMacro	817
25.322.3.33tkGetMacro	817
25.322.3.34tkGetMacro	817
25.322.3.35tkGetObjectMacro	817
25.322.3.36tkGetObjectMacro	817
25.322.3.37tkGetObjectMacro	817
25.322.3.38tkGetObjectMacro	817
25.322.3.39tkGetStringMacro	817
25.322.3.40tkGetStringMacro	817
25.322.3.41tkGetVector3Macro	817
25.322.3.42tkGetVector6Macro	817
25.322.3.43tkSetMacro	817
25.322.3.44tkSetMacro	817
25.322.3.45tkSetMacro	818
25.322.3.46tkSetMacro	818
25.322.3.47tkSetVector6Macro	818
25.322.3.48tkTypeRevisionMacro	818
25.322.4 Member Data Documentation	818
25.322.4.1ApplyInverseVideo	818
25.322.4.2ApplyLookupTable	818
25.322.4.3ApplyPlanarConfiguration	818
25.322.4.4ApplyShiftScale	818
25.322.4.5ApplyYBRToRGB	818
25.322.4.6Curve	818
25.322.4.7DirectionCosines	818
25.322.4.8FileNames	818

25.322.4.9ForceRescale	818
25.322.4.10IconDataScalarType	818
25.322.4.11IconImageDataExtent	818
25.322.4.12IconNumberOfScalarComponents	818
25.322.4.13ImageFormat	818
25.322.4.14ImageOrientationPatient	818
25.322.4.15ImagePositionPatient	818
25.322.4.16LoadIconImage	818
25.322.4.17LoadOverlays	818
25.322.4.18LossyFlag	818
25.322.4.19MedicalImageProperties	818
25.322.4.20NumberOfIconImages	818
25.322.4.21NumberOfOverlays	818
25.322.4.22PlanarConfiguration	819
25.322.4.23Scale	819
25.322.4.24Shift	819
25.323.vtkGDCMImageWriter Class Reference	819
25.323.1Detailed Description	820
25.323.2Member Enumeration Documentation	820
25.323.2.1CompressionTypes	820
25.323.3Constructor & Destructor Documentation	820
25.323.3.1vtkGDCMImageWriter	820
25.323.3.2~vtkGDCMImageWriter	820
25.323.4Member Function Documentation	820
25.323.4.1GetDescriptiveName	821
25.323.4.2GetFileExtensions	821
25.323.4.3GetFileName	821
25.323.4.4New	821
25.323.4.5PrintSelf	821
25.323.4.6SetDirectionCosines	821
25.323.4.7SetDirectionCosinesFromImageOrientationPatient	821
25.323.4.8SetFileNames	821
25.323.4.9SetMedicalImageProperties	821
25.323.4.10kBooleanMacro	821
25.323.4.11kBooleanMacro	821
25.323.4.12kGetMacro	821
25.323.4.13kGetMacro	821

25.323.4.1	vtkGetMacro	821
25.323.4.1	vtkGetMacro	822
25.323.4.1	vtkGetMacro	822
25.323.4.1	vtkGetMacro	822
25.323.4.1	vtkGetMacro	822
25.323.4.1	vtkGetObjectMacro	822
25.323.4.2	vtkGetObjectMacro	822
25.323.4.2	vtkGetObjectMacro	822
25.323.4.2	vtkGetStringMacro	822
25.323.4.2	vtkGetStringMacro	822
25.323.4.2	vtkSetMacro	822
25.323.4.2	vtkSetMacro	822
25.323.4.2	vtkSetMacro	822
25.323.4.2	vtkSetMacro	822
25.323.4.2	vtkSetMacro	822
25.323.4.2	vtkSetMacro	822
25.323.4.2	vtkSetMacro	822
25.323.4.3	vtkSetStringMacro	822
25.323.4.3	vtkSetStringMacro	822
25.323.4.3	vtkTypeRevisionMacro	822
25.323.4.3	Write	822
25.323.4.3	WriteGDCMData	822
25.323.4.3	WriteSlice	822
25.324	vtkGDCMMedicalImageProperties Class Reference	823
25.324.1	Constructor & Destructor Documentation	823
25.324.1.1	vtkGDCMMedicalImageProperties	823
25.324.1.2	~vtkGDCMMedicalImageProperties	823
25.324.2	Member Function Documentation	823
25.324.2.1	Clear	823
25.324.2.2	GetFile	823
25.324.2.3	New	823
25.324.2.4	PrintSelf	823
25.324.2.5	PushBackFile	823
25.324.2.6	vtkTypeRevisionMacro	824
25.324.3	Friends And Related Function Documentation	824
25.324.3.1	vtkGDCMImageReader	824
25.324.3.2	vtkGDCMImageWriter	824

25.325.1	vtkGDCMPolyDataReader Class Reference	824
25.325.1	Detailed Description	825
25.325.2	Constructor & Destructor Documentation	825
25.325.2.1	vtkGDCMPolyDataReader	825
25.325.2.2	~vtkGDCMPolyDataReader	825
25.325.3	Member Function Documentation	825
25.325.3.1	FillMedicalImageInformation	825
25.325.3.2	New	825
25.325.3.3	PrintSelf	825
25.325.3.4	RequestData	825
25.325.3.5	RequestData_HemodynamicWaveformStorage	825
25.325.3.6	RequestData_RTStructureSetStorage	826
25.325.3.7	RequestInformation	826
25.325.3.8	RequestInformation_HemodynamicWaveformStorage	826
25.325.3.9	RequestInformation_RTStructureSetStorage	826
25.325.3.10	GetObjectMacro	826
25.325.3.11	GetObjectMacro	826
25.325.3.12	GetStringMacro	826
25.325.3.13	SetStringMacro	826
25.325.3.14	TypeRevisionMacro	826
25.325.4	Member Data Documentation	826
25.325.4.1	FileName	826
25.325.4.2	MedicalImageProperties	826
25.325.4.3	RTStructSetProperties	826
25.326.1	vtkGDCMPolyDataWriter Class Reference	826
25.326.1	Detailed Description	827
25.326.2	Constructor & Destructor Documentation	828
25.326.2.1	vtkGDCMPolyDataWriter	828
25.326.2.2	~vtkGDCMPolyDataWriter	828
25.326.3	Member Function Documentation	828
25.326.3.1	InitializeRTStructSet	828
25.326.3.2	New	828
25.326.3.3	PrintSelf	828
25.326.3.4	SetMedicalImageProperties	828
25.326.3.5	SetNumberOfInputPorts	828
25.326.3.6	SetRTStructSetProperties	828
25.326.3.7	TypeRevisionMacro	828

25.326.3.8WriteData	828
25.326.3.9WriteRTSTRUCTData	828
25.326.3.10WriteRTSTRUCTInfo	828
25.326.4Member Data Documentation	829
25.326.4.1MedicalImageProperties	829
25.326.4.2RTStructSetProperties	829
25.327.1GDCMTesting Class Reference	829
25.327.1Detailed Description	829
25.327.2Member Typedef Documentation	830
25.327.2.1MD5MetalmagesType	830
25.327.3Constructor & Destructor Documentation	830
25.327.3.1vtkGDCMTesting	830
25.327.3.2~vtkGDCMTesting	830
25.327.4Member Function Documentation	830
25.327.4.1GetGDCMDataRoot	830
25.327.4.2GetMD5Metalmage	830
25.327.4.3GetMHDMD5FromFile	830
25.327.4.4GetNumberOfMD5Metalmages	830
25.327.4.5GetRAWMD5FromFile	830
25.327.4.6GetVTKDataRoot	830
25.327.4.7New	830
25.327.4.8PrintSelf	831
25.327.4.9vtkTypeRevisionMacro	831
25.328.1GDCMThreadedImageReader Class Reference	831
25.328.1Constructor & Destructor Documentation	832
25.328.1.1vtkGDCMThreadedImageReader	832
25.328.1.2~vtkGDCMThreadedImageReader	832
25.328.2Member Function Documentation	832
25.328.2.1ExecuteData	832
25.328.2.2ExecuteInformation	832
25.328.2.3New	832
25.328.2.4PrintSelf	832
25.328.2.5ReadFiles	833
25.328.2.6RequestDataCompat	833
25.328.2.7vtkBooleanMacro	833
25.328.2.8vtkGetMacro	833
25.328.2.9vtkSetMacro	833

25.328.2.1	vtkSetMacro	833
25.328.2.1	vtkSetMacro	833
25.328.2.1	vtkTypeRevisionMacro	833
25.329	vtkGDCMThreadedImageReader2 Class Reference	833
25.329.1	Constructor & Destructor Documentation	834
25.329.1.1	vtkGDCMThreadedImageReader2	834
25.329.1.2	~vtkGDCMThreadedImageReader2	834
25.329.2	Member Function Documentation	834
25.329.2.1	GetFileName	834
25.329.2.2	New	834
25.329.2.3	PrintSelf	834
25.329.2.4	RequestInformation	834
25.329.2.5	SetFileName	834
25.329.2.6	SetFileNames	834
25.329.2.7	SplitExtent	834
25.329.2.8	ThreadedRequestData	835
25.329.2.9	vtkBooleanMacro	835
25.329.2.10	vtkBooleanMacro	835
25.329.2.11	vtkBooleanMacro	835
25.329.2.12	vtkGetMacro	835
25.329.2.13	vtkGetMacro	835
25.329.2.14	vtkGetMacro	835
25.329.2.15	vtkGetMacro	835
25.329.2.16	vtkGetMacro	835
25.329.2.17	vtkGetMacro	835
25.329.2.18	vtkGetMacro	835
25.329.2.19	vtkGetMacro	835
25.329.2.20	vtkGetObjectMacro	835
25.329.2.21	vtkGetVector3Macro	835
25.329.2.22	vtkGetVector3Macro	835
25.329.2.23	vtkGetVector6Macro	835
25.329.2.24	vtkSetMacro	835
25.329.2.25	vtkSetMacro	835
25.329.2.26	vtkSetMacro	835
25.329.2.27	vtkSetMacro	835
25.329.2.28	vtkSetMacro	835
25.329.2.29	vtkSetMacro	835

25.329.2.30	vtkSetMacro	835
25.329.2.31	vtkSetVector3Macro	835
25.329.2.32	vtkSetVector3Macro	835
25.329.2.33	vtkSetVector6Macro	836
25.329.2.34	vtkTypeRevisionMacro	836
25.330	vtkImageColorViewer Class Reference	836
25.330.1	Detailed Description	838
25.330.2	Member Enumeration Documentation	838
25.330.2.1	anonymous enum	838
25.330.3	Constructor & Destructor Documentation	838
25.330.3.1	vtkImageColorViewer	838
25.330.3.2	~vtkImageColorViewer	838
25.330.4	Member Function Documentation	838
25.330.4.1	AddInput	838
25.330.4.2	AddInputConnection	838
25.330.4.3	GetColorLevel	838
25.330.4.4	GetColorWindow	839
25.330.4.5	GetInput	839
25.330.4.6	GetOffScreenRendering	839
25.330.4.7	GetOverlayVisibility	839
25.330.4.8	GetPosition	839
25.330.4.9	GetSize	839
25.330.4.10	GetSliceMax	839
25.330.4.11	GetSliceMin	839
25.330.4.12	GetSliceRange	839
25.330.4.13	GetSliceRange	839
25.330.4.14	GetSliceRange	839
25.330.4.15	GetWindowName	839
25.330.4.16	InstallPipeline	839
25.330.4.17	New	839
25.330.4.18	PrintSelf	839
25.330.4.19	Render	839
25.330.4.20	SetColorLevel	839
25.330.4.21	SetColorWindow	839
25.330.4.22	SetDisplayId	839
25.330.4.23	SetInput	840
25.330.4.24	SetInputConnection	840

25.330.4.25	SetOffScreenRendering	840
25.330.4.26	SetOverlayVisibility	840
25.330.4.27	SetParentId	840
25.330.4.28	SetPosition	840
25.330.4.29	SetPosition	840
25.330.4.30	SetRenderer	840
25.330.4.31	SetRenderWindow	840
25.330.4.32	SetSize	840
25.330.4.33	SetSize	840
25.330.4.34	SetSlice	840
25.330.4.35	SetSliceOrientation	840
25.330.4.36	SetSliceOrientationToXY	840
25.330.4.37	SetSliceOrientationToXZ	840
25.330.4.38	SetSliceOrientationToYZ	841
25.330.4.39	SetupInteractor	841
25.330.4.40	SetWindowId	841
25.330.4.41	InstallPipeline	841
25.330.4.42	UpdateDisplayExtent	841
25.330.4.43	UpdateOrientation	841
25.330.4.44	TK_LEGACY	841
25.330.4.45	TK_LEGACY	841
25.330.4.46	TK_LEGACY	841
25.330.4.47	TK_LEGACY	841
25.330.4.48	BooleanMacro	841
25.330.4.49	GetMacro	841
25.330.4.50	GetMacro	841
25.330.4.51	GetObjectMacro	841
25.330.4.52	GetObjectMacro	841
25.330.4.53	GetObjectMacro	841
25.330.4.54	GetObjectMacro	841
25.330.4.55	GetObjectMacro	841
25.330.4.56	TypeRevisionMacro	841
25.330.5	Member Data Documentation	841
25.330.5.1	FirstRender	841
25.330.5.2	ImageActor	841
25.330.5.3	Interactor	842
25.330.5.4	InteractorStyle	842

25.330.5.5OverlayImageActor	842
25.330.5.6Renderer	842
25.330.5.7RenderWindow	842
25.330.5.8Slice	842
25.330.5.9SliceOrientation	842
25.330.5.10WindowLevel	842
25.331.vtkImageMapToColors16 Class Reference	842
25.331.1.Constructor & Destructor Documentation	843
25.331.1.1vtkImageMapToColors16	843
25.331.1.2~vtkImageMapToColors16	843
25.331.2.Member Function Documentation	843
25.331.2.1GetMTime	843
25.331.2.2New	843
25.331.2.3PrintSelf	843
25.331.2.4RequestData	843
25.331.2.5RequestInformation	843
25.331.2.6SetLookupTable	843
25.331.2.7SetOutputFormatToLuminance	843
25.331.2.8SetOutputFormatToLuminanceAlpha	843
25.331.2.9SetOutputFormatToRGB	843
25.331.2.10SetOutputFormatToRGBA	843
25.331.2.11ThreadedRequestData	843
25.331.2.12vtkBooleanMacro	844
25.331.2.13vtkGetMacro	844
25.331.2.14vtkGetMacro	844
25.331.2.15vtkGetMacro	844
25.331.2.16vtkGetObjectMacro	844
25.331.2.17vtkSetMacro	844
25.331.2.18vtkSetMacro	844
25.331.2.19vtkSetMacro	844
25.331.2.20vtkTypeRevisionMacro	844
25.331.3.Member Data Documentation	844
25.331.3.1ActiveComponent	844
25.331.3.2DataWasPassed	844
25.331.3.3LookupTable	844
25.331.3.4OutputFormat	844
25.331.3.5PassAlphaToOutput	844

25.332	vtkImageMapToWindowLevelColors2 Class Reference	844
25.332.1	Constructor & Destructor Documentation	845
25.332.1.1	vtkImageMapToWindowLevelColors2	845
25.332.1.2	~vtkImageMapToWindowLevelColors2	845
25.332.2	Member Function Documentation	845
25.332.2.1	New	845
25.332.2.2	PrintSelf	845
25.332.2.3	RequestData	845
25.332.2.4	RequestInformation	845
25.332.2.5	ThreadedRequestData	845
25.332.2.6	vtkGetMacro	845
25.332.2.7	vtkGetMacro	845
25.332.2.8	vtkSetMacro	845
25.332.2.9	vtkSetMacro	845
25.332.2.10	vtkTypeRevisionMacro	846
25.332.3	Member Data Documentation	846
25.332.3.1	Level	846
25.332.3.2	Window	846
25.333	vtkImagePlanarComponentsToComponents Class Reference	846
25.333.1	Constructor & Destructor Documentation	846
25.333.1.1	vtkImagePlanarComponentsToComponents	846
25.333.1.2	~vtkImagePlanarComponentsToComponents	846
25.333.2	Member Function Documentation	846
25.333.2.1	New	846
25.333.2.2	PrintSelf	846
25.333.2.3	RequestData	847
25.333.2.4	vtkTypeRevisionMacro	847
25.334	vtkImageRGBToYBR Class Reference	847
25.334.1	Constructor & Destructor Documentation	847
25.334.1.1	vtkImageRGBToYBR	847
25.334.1.2	~vtkImageRGBToYBR	847
25.334.2	Member Function Documentation	847
25.334.2.1	New	847
25.334.2.2	PrintSelf	847
25.334.2.3	ThreadedExecute	847
25.334.2.4	vtkTypeRevisionMacro	847
25.335	vtkImageYBRToRGB Class Reference	848

25.335.1	Constructor & Destructor Documentation	848
25.335.1.1	vtkImageYBRToRGB	848
25.335.1.2	~vtkImageYBRToRGB	848
25.335.2	Member Function Documentation	848
25.335.2.1	New	848
25.335.2.2	PrintSelf	848
25.335.2.3	ThreadedExecute	848
25.335.2.4	vtkTypeRevisionMacro	848
25.336	vtkLookupTable16 Class Reference	848
25.336.1	Constructor & Destructor Documentation	849
25.336.1.1	vtkLookupTable16	849
25.336.1.2	~vtkLookupTable16	849
25.336.2	Member Function Documentation	849
25.336.2.1	Build	849
25.336.2.2	GetPointer	849
25.336.2.3	MapScalarsThroughTable2	849
25.336.2.4	New	849
25.336.2.5	PrintSelf	849
25.336.2.6	SetNumberOfTableValues	849
25.336.2.7	vtkTypeRevisionMacro	849
25.336.2.8	WritePointer	849
25.336.3	Member Data Documentation	850
25.336.3.1	Table16	850
25.337	vtkRTStructSetProperties Class Reference	850
25.337.1	Detailed Description	851
25.337.2	Constructor & Destructor Documentation	851
25.337.2.1	vtkRTStructSetProperties	851
25.337.2.2	~vtkRTStructSetProperties	851
25.337.3	Member Function Documentation	851
25.337.3.1	AddContourReferencedFrameOfReference	851
25.337.3.2	AddReferencedFrameOfReference	851
25.337.3.3	AddStructureSetROI	852
25.337.3.4	AddStructureSetROIObservation	852
25.337.3.5	Clear	852
25.337.3.6	DeepCopy	852
25.337.3.7	GetContourReferencedFrameOfReferenceClassUID	852
25.337.3.8	GetContourReferencedFrameOfReferenceInstanceUID	852

25.337.3.9	GetNumberOfContourReferencedFrameOfReferences	852
25.337.3.10	GetNumberOfContourReferencedFrameOfReferences	852
25.337.3.10	GetNumberOfReferencedFrameOfReferences	852
25.337.3.10	GetNumberOfStructureSetROIs	852
25.337.3.10	GetReferencedFrameOfReferenceClassUID	852
25.337.3.10	GetReferencedFrameOfReferenceInstanceUID	852
25.337.3.10	GetStructureSetObservationNumber	852
25.337.3.10	GetStructureSetROIGenerationAlgorithm	852
25.337.3.10	GetStructureSetROIName	852
25.337.3.10	GetStructureSetROINumber	852
25.337.3.10	GetStructureSetROIRefFrameRefUID	852
25.337.3.20	GetStructureSetRTROIInterpretedType	852
25.337.3.21	New	852
25.337.3.22	PrintSelf	852
25.337.3.23	GetStringMacro	852
25.337.3.24	GetStringMacro	852
25.337.3.25	GetStringMacro	853
25.337.3.26	GetStringMacro	853
25.337.3.27	GetStringMacro	853
25.337.3.28	GetStringMacro	853
25.337.3.29	GetStringMacro	853
25.337.3.30	GetStringMacro	853
25.337.3.31	GetStringMacro	853
25.337.3.32	SetStringMacro	853
25.337.3.33	SetStringMacro	853
25.337.3.34	SetStringMacro	853
25.337.3.35	SetStringMacro	853
25.337.3.36	SetStringMacro	853
25.337.3.37	SetStringMacro	853
25.337.3.38	SetStringMacro	853
25.337.3.39	SetStringMacro	853
25.337.3.40	SetStringMacro	853
25.337.3.41	TypeRevisionMacro	853
25.337.4	Member Data Documentation	853
25.337.4.1	Internals	853
25.337.4.2	ReferenceFrameOfReferenceUID	853
25.337.4.3	ReferenceSeriesInstanceUID	853

25.337.4.4SeriesInstanceUID	853
25.337.4.5SOPInstanceUID	853
25.337.4.6StructureSetDate	853
25.337.4.7StructureSetLabel	853
25.337.4.8StructureSetName	853
25.337.4.9StructureSetTime	854
25.337.4.10StudyInstanceUID	854
25.338dcm::Waveform Class Reference	854
25.338.1Detailed Description	854
25.338.2Constructor & Destructor Documentation	854
25.338.2.1Waveform	854
25.339dcm::Writer Class Reference	854
25.339.1Detailed Description	857
25.339.2Constructor & Destructor Documentation	858
25.339.2.1Writer	858
25.339.2.2~Writer	858
25.339.3Member Function Documentation	858
25.339.3.1CheckFileMetaInformationOff	858
25.339.3.2CheckFileMetaInformationOn	858
25.339.3.3GetFile	858
25.339.3.4GetStreamPtr	858
25.339.3.5SetCheckFileMetaInformation	858
25.339.3.6SetFile	858
25.339.3.7SetFileName	859
25.339.3.8SetStream	859
25.339.3.9SetWriteDataSetOnly	859
25.339.3.10Write	859
25.339.4Friends And Related Function Documentation	859
25.339.4.1StreamImageWriter	859
25.339.5Member Data Documentation	859
25.339.5.1ofstream	859
25.339.5.2Stream	859
25.340dcm::XMLDictReader Class Reference	860
25.340.1Detailed Description	861
25.340.2Constructor & Destructor Documentation	861
25.340.2.1XMLDictReader	861
25.340.2.2~XMLDictReader	861

25.340.3	Member Function Documentation	861
25.340.3.1	CharacterDataHandler	861
25.340.3.2	EndElement	861
25.340.3.3	GetDict	861
25.340.3.4	HandleDescription	861
25.340.3.5	HandleEntry	861
25.340.3.6	StartElement	861
25.341	gdcm::XMLPrivateDictReader Class Reference	861
25.341.1	Detailed Description	862
25.341.2	Constructor & Destructor Documentation	863
25.341.2.1	XMLPrivateDictReader	863
25.341.2.2	~XMLPrivateDictReader	863
25.341.3	Member Function Documentation	863
25.341.3.1	CharacterDataHandler	863
25.341.3.2	EndElement	863
25.341.3.3	GetPrivateDict	863
25.341.3.4	HandleDescription	863
25.341.3.5	HandleEntry	863
25.341.3.6	StartElement	863
26	File Documentation	865
26.1	gdcm2pnm.man File Reference	865
26.2	gdcm2vtk.man File Reference	865
26.3	gdcmAAbortPDU.h File Reference	865
26.4	gdcmAAAssociateACPDU.h File Reference	865
26.5	gdcmAAAssociateRJPDU.h File Reference	866
26.6	gdcmAAAssociateRQPDU.h File Reference	866
26.7	gdcmAbstractSyntax.h File Reference	866
26.8	gdcmanon.man File Reference	866
26.9	gdcmAnonymizeEvent.h File Reference	866
26.10	gdcmAnonymizer.h File Reference	867
26.11	gdcmApplicationContext.h File Reference	867
26.12	gdcmApplicationEntity.h File Reference	867
26.13	gdcmAReleaseRPPDU.h File Reference	867
26.14	gdcmAReleaseRQPDU.h File Reference	868
26.15	gdcmARTIMTimer.h File Reference	868
26.16	gdcmASN1.h File Reference	868

26.17gdcmlAsynchronousOperationsWindowSub.h File Reference	869
26.18gdcmlAttribute.h File Reference	869
26.19gdcmlAudioCodec.h File Reference	869
26.20gdcmlBase64.h File Reference	870
26.21gdcmlBaseCompositeMessage.h File Reference	870
26.22gdcmlBasePDU.h File Reference	870
26.23gdcmlBaseRootQuery.h File Reference	870
26.24gdcmlBasicOffsetTable.h File Reference	871
26.25gdcmlBitmap.h File Reference	871
26.26gdcmlBitmapToBitmapFilter.h File Reference	872
26.27gdcmlByteBuffer.h File Reference	872
26.28gdcmlByteSwap.h File Reference	872
26.29gdcmlByteSwapFilter.h File Reference	872
26.30gdcmlByteValue.h File Reference	873
26.31gdcmlCEchoMessages.h File Reference	873
26.32gdcmlCFindMessages.h File Reference	873
26.33gdcmlCMoveMessages.h File Reference	873
26.34gdcmlCodec.h File Reference	874
26.35gdcmlCoder.h File Reference	874
26.36gdcmlCodeString.h File Reference	874
26.37gdcmlCommand.h File Reference	875
26.38gdcmlCommandDataSet.h File Reference	875
26.39gdcmlCompositeMessageFactory.h File Reference	875
26.40gdcmlCompositeNetworkFunctions.h File Reference	876
26.41gdcmlConstCharWrapper.h File Reference	876
26.42gdcmlconv.man File Reference	876
26.43gdcmlCP246ExplicitDataElement.h File Reference	876
26.44gdcmlCryptographicMessageSyntax.h File Reference	876
26.45gdcmlCSAElement.h File Reference	877
26.46gdcmlCSAHeader.h File Reference	877
26.47gdcmlCSAHeaderDict.h File Reference	877
26.48gdcmlCSAHeaderDictEntry.h File Reference	878
26.49gdcmlCStoreMessages.h File Reference	878
26.50gdcmlCurve.h File Reference	878
26.51gdcmlDataElement.h File Reference	879
26.52gdcmlDataEvent.h File Reference	879
26.53gdcmlDataSet.h File Reference	879

26.54gdcmDataSetEvent.h File Reference	880
26.55gdcmDataSetHelper.h File Reference	880
26.56gdcmDecoder.h File Reference	880
26.57gdcmDefinedTerms.h File Reference	880
26.58gdcmDeflateStream.h File Reference	881
26.59gdcmDefs.h File Reference	881
26.60gdcmDeltaEncodingCodec.h File Reference	881
26.61gdcmDICOMDIR.h File Reference	881
26.62gdcmDICOMDIRGenerator.h File Reference	882
26.63gdcmDict.h File Reference	882
26.64gdcmDictConverter.h File Reference	882
26.65gdcmDictEntry.h File Reference	883
26.66gdcmDictPrinter.h File Reference	883
26.67gdcmDicts.h File Reference	883
26.68gdcmdiff.man File Reference	884
26.69gdcmDIMSE.h File Reference	884
26.70gdcmDirectionCosines.h File Reference	884
26.71gdcmDirectory.h File Reference	884
26.72gdcmDirectoryHelper.h File Reference	885
26.73gdcmDummyValueGenerator.h File Reference	885
26.74gdcmdump.man File Reference	885
26.75gdcmDumper.h File Reference	885
26.76gdcmElement.h File Reference	885
26.77gdcmEncapsulatedDocument.h File Reference	886
26.78gdcmEnumeratedValues.h File Reference	886
26.79gdcmEvent.h File Reference	887
26.79.1 Macro Definition Documentation	887
26.79.1.1 gdcmEventMacro	887
26.80gdcmException.h File Reference	888
26.81gdcmExplicitDataElement.h File Reference	888
26.82gdcmExplicitImplicitDataElement.h File Reference	888
26.83gdcmFiducials.h File Reference	888
26.84gdcmFile.h File Reference	889
26.85gdcmFileDerivation.h File Reference	889
26.86gdcmFileExplicitFilter.h File Reference	889
26.87gdcmFileMetaInformation.h File Reference	890
26.88gdcmFilename.h File Reference	890

26.89gdcmlFilenameGenerator.h File Reference	890
26.90gdcmlFileSet.h File Reference	890
26.91gdcmlFindPatientRootQuery.h File Reference	891
26.92gdcmlFindStudyRootQuery.h File Reference	891
26.93gdcmlFragment.h File Reference	891
26.94gdcmlgendir.man File Reference	892
26.95gdcmlGlobal.h File Reference	892
26.96gdcmlGroupDict.h File Reference	892
26.97gdcmlIconImage.h File Reference	892
26.98gdcmlIconImageFilter.h File Reference	893
26.99gdcmlIconImageGenerator.h File Reference	893
26.100gdcmlImage.h File Reference	893
26.101gdcmlImageApplyLookupTable.h File Reference	894
26.102gdcmlImageChangePhotometricInterpretation.h File Reference	894
26.103gdcmlImageChangePlanarConfiguration.h File Reference	894
26.104gdcmlImageChangeTransferSyntax.h File Reference	894
26.105gdcmlImageCodec.h File Reference	895
26.106gdcmlImageConverter.h File Reference	895
26.107gdcmlImageFragmentSplitter.h File Reference	895
26.108gdcmlImageHelper.h File Reference	895
26.109gdcmlImageReader.h File Reference	896
26.110gdcmlImageToImageFilter.h File Reference	896
26.111gdcmlImageWriter.h File Reference	896
26.112gdcmlimg.man File Reference	896
26.113gdcmlImplementationClassUIDSub.h File Reference	896
26.114gdcmlImplementationUIDSub.h File Reference	897
26.115gdcmlImplementationVersionNameSub.h File Reference	897
26.116gdcmlImplicitDataElement.h File Reference	897
26.117gdcmlinfo.man File Reference	898
26.118gdcmlIOD.h File Reference	898
26.119gdcmlIODEntry.h File Reference	898
26.120gdcmlIODs.h File Reference	898
26.121gdcmlIPPSorter.h File Reference	899
26.122gdcmlItem.h File Reference	899
26.123gdcmlJPEG12Codec.h File Reference	899
26.124gdcmlJPEG16Codec.h File Reference	900
26.125gdcmlJPEG2000Codec.h File Reference	900

26.126	dcmJPEG8Codec.h File Reference	900
26.127	dcmJPEGCodec.h File Reference	900
26.128	dcmJPEGLSCodec.h File Reference	901
26.129	dcmKAKADUCodec.h File Reference	901
26.130	dcmLegacyMacro.h File Reference	901
26.130.1	Macro Definition Documentation	901
26.130.1.1	1GDCM_LEGACY	901
26.130.1.2	2GDCM_LEGACY_BODY	901
26.130.1.3	3GDCM_LEGACY_REPLACED_BODY	901
26.131	dcmLO.h File Reference	901
26.132	dcmLookupTable.h File Reference	902
26.133	dcmMacro.h File Reference	902
26.134	dcmMacroEntry.h File Reference	902
26.134.1	Macro Definition Documentation	902
26.134.1.1	1GDCMMACROENTRY_H	902
26.135	dcmMacros.h File Reference	902
26.136	dcmMaximumLengthSub.h File Reference	903
26.137	dcmMD5.h File Reference	903
26.138	dcmMediaStorage.h File Reference	903
26.139	dcmMeshPrimitive.h File Reference	904
26.140	dcmModule.h File Reference	904
26.141	dcmModuleEntry.h File Reference	904
26.142	dcmModules.h File Reference	905
26.143	dcmMovePatientRootQuery.h File Reference	905
26.144	dcmMoveStudyRootQuery.h File Reference	905
26.145	dcmNestedModuleEntries.h File Reference	906
26.146	dcmNetworkEvents.h File Reference	906
26.147	dcmNetworkStateID.h File Reference	907
26.148	dcmObject.h File Reference	907
26.149	dcmOrientation.h File Reference	908
26.150	dcmOverlay.h File Reference	908
26.151	dcmParseException.h File Reference	908
26.152	dcmParser.h File Reference	909
26.153	dcmPatient.h File Reference	909
26.154	dcmPDataTFPDU.h File Reference	909
26.155	dcmPDBElement.h File Reference	909
26.156	dcmPDBHeader.h File Reference	910

26.157	dcmpdf.man File Reference	910
26.158	dcmPDFCodec.h File Reference	910
26.159	dcmPDUFactory.h File Reference	910
26.160	dcmPersonName.h File Reference	911
26.161	dcmPhotometricInterpretation.h File Reference	911
26.162	dcmPixelFormat.h File Reference	911
26.163	dcmPixmap.h File Reference	912
26.164	dcmPixmapReader.h File Reference	912
26.165	dcmPixmapToPixmapFilter.h File Reference	912
26.166	dcmPixmapWriter.h File Reference	912
26.167	dcmPNMCodec.h File Reference	913
26.168	dcmPreamble.h File Reference	913
26.169	dcmPresentationContext.h File Reference	913
26.170	dcmPresentationContextAC.h File Reference	913
26.171	dcmPresentationContextGenerator.h File Reference	914
26.172	dcmPresentationContextRQ.h File Reference	914
26.173	dcmPresentationDataValue.h File Reference	914
26.174	dcmPrinter.h File Reference	915
26.175	dcmPrivateTag.h File Reference	915
26.176	dcmProgressEvent.h File Reference	915
26.177	dcmPVRGCodec.h File Reference	915
26.178	dcmPythonFilter.h File Reference	916
26.179	dcmQueryBase.h File Reference	916
26.180	dcmQueryFactory.h File Reference	916
26.181	dcmQueryImage.h File Reference	917
26.182	dcmQueryPatient.h File Reference	917
26.183	dcmQuerySeries.h File Reference	917
26.184	dcmQueryStudy.h File Reference	918
26.185	dcmraw.man File Reference	918
26.186	dcmRAWCodec.h File Reference	918
26.187	dcmReader.h File Reference	918
26.188	dcmRescaler.h File Reference	919
26.189	dcmRLECodec.h File Reference	919
26.190	dcmScanner.h File Reference	919
26.191	dcmscanner.man File Reference	920
26.192	dcmscu.man File Reference	920
26.193	dcmSegment.h File Reference	920

26.194	dcmSegmentedPaletteColorLookupTable.h File Reference	920
26.195	dcmSegmentHelper.h File Reference	920
26.196	dcmSegmentReader.h File Reference	920
26.197	dcmSegmentWriter.h File Reference	921
26.198	dcmSequenceOfFragments.h File Reference	921
26.199	dcmSequenceOfItems.h File Reference	921
26.200	dcmSerieHelper.h File Reference	921
26.201	dcmSeries.h File Reference	922
26.202	dcmServiceClassUser.h File Reference	922
26.203	dcmSHA1.h File Reference	923
26.204	dcmSimpleSubjectWatcher.h File Reference	923
26.205	dcmSmartPointer.h File Reference	923
26.206	dcmSOPClassUIDToIOD.h File Reference	923
26.207	dcmSorter.h File Reference	924
26.208	dcmSpacing.h File Reference	924
26.209	dcmSpectroscopy.h File Reference	924
26.210	dcmSplitMosaicFilter.h File Reference	924
26.211	dcmStaticAssert.h File Reference	925
26.211.1	Macro Definition Documentation	925
26.211.1.1	GDGM_DO_JOIN	925
26.211.1.2	GDGM_DO_JOIN2	925
26.211.1.3	GDGM_JOIN	925
26.211.1.4	GDGM_STATIC_ASSERT	925
26.212	dcmStreamImageReader.h File Reference	925
26.213	dcmStreamImageWriter.h File Reference	926
26.214	dcmString.h File Reference	926
26.215	dcmStringFilter.h File Reference	926
26.216	dcmStudy.h File Reference	927
26.217	dcmSubject.h File Reference	927
26.218	dcmSurface.h File Reference	927
26.219	dcmSurfaceHelper.h File Reference	927
26.220	dcmSurfaceReader.h File Reference	928
26.221	dcmSurfaceWriter.h File Reference	928
26.222	dcmSwapCode.h File Reference	928
26.223	dcmSwapper.h File Reference	928
26.224	dcmSystem.h File Reference	929
26.225	dcmTable.h File Reference	929

26.226	dcmTableEntry.h File Reference	929
26.227	dcmTableReader.h File Reference	929
26.228	dcmTag.h File Reference	930
26.229	dcmTagPath.h File Reference	930
26.230	dcmTagToVR.h File Reference	930
26.231	dcmTar.man File Reference	931
26.232	dcmTerminal.h File Reference	931
26.233	dcmTestDriver.h File Reference	931
26.234	dcmTesting.h File Reference	931
26.235	dcmTrace.h File Reference	932
26.235.1	Macro Definition Documentation	932
26.235.1.1	GDCM_FUNCTION	932
26.235.1.2	dcmAssertAlwaysMacro	932
26.235.1.3	dcmAssertMacro	932
26.235.1.4	dcmDebugMacro	933
26.235.1.5	dcmErrorMacro	933
26.235.1.6	dcmWarningMacro	934
26.236	dcmTransferSyntax.h File Reference	934
26.237	dcmTransferSyntaxSub.h File Reference	935
26.238	dcmType.h File Reference	935
26.239	dcmTypes.h File Reference	935
26.239.1	Macro Definition Documentation	935
26.239.1.1	UINT32_MAX	935
26.240	dcmUIDGenerator.h File Reference	935
26.241	dcmUIDs.h File Reference	936
26.242	dcmULAction.h File Reference	936
26.243	dcmULActionAA.h File Reference	936
26.244	dcmULActionAE.h File Reference	937
26.245	dcmULActionAR.h File Reference	937
26.246	dcmULActionDT.h File Reference	938
26.247	dcmULBasicCallback.h File Reference	938
26.248	dcmULConnection.h File Reference	938
26.249	dcmULConnectionCallback.h File Reference	938
26.250	dcmULConnectionInfo.h File Reference	939
26.251	dcmULConnectionManager.h File Reference	939
26.252	dcmULEvent.h File Reference	939
26.253	dcmULTransitionTable.h File Reference	939

26.254	dcmULWritingCallback.h File Reference	940
26.255	dcmUNExplicitDataElement.h File Reference	940
26.256	dcmUNExplicitImplicitDataElement.h File Reference	940
26.257	dcmUnpacker12Bits.h File Reference	941
26.258	dcmUsage.h File Reference	941
26.259	dcmUserInformation.h File Reference	941
26.260	dcmValidate.h File Reference	941
26.261	dcmValue.h File Reference	942
26.262	dcmValueIO.h File Reference	942
26.263	dcmVersion.h File Reference	942
26.264	dcmviewer.man File Reference	943
26.265	dcmVL.h File Reference	943
26.266	dcmVM.h File Reference	943
26.266.1	Macro Definition Documentation	943
26.266.1.1	TYPETOLENGTH	943
26.267	dcmVR.h File Reference	944
26.267.1	Macro Definition Documentation	945
26.267.1.1	TYPETOENCODING	945
26.267.1.2	VRTypeTemplateCase	945
26.268	dcmVR16ExplicitDataElement.h File Reference	945
26.269	dcmWaveform.h File Reference	945
26.270	dcmWin32.h File Reference	945
26.270.1	Macro Definition Documentation	946
26.270.1.1	GDCM_EXPORT	946
26.271	dcmWriter.h File Reference	946
26.272	dcmXMLDictReader.h File Reference	946
26.273	dcmXMLPrivateDictReader.h File Reference	946
26.274	tkGDCMImageIO2.h File Reference	946
26.274.1	Macro Definition Documentation	947
26.274.1.1	ITK_GDCM_EXPORT	947
26.275	README.txt File Reference	947
26.276	TestsList.txt File Reference	947
26.277	tkGDCMImageReader.h File Reference	947
26.277.1	Macro Definition Documentation	947
26.277.1.1	VTK_CMYK	947
26.277.1.2	VTK_INVERSE_LUMINANCE	947
26.277.1.3	VTK_LOOKUP_TABLE	947

26.277.1.4VTK_YBR	947
26.278tkGDCMImageWriter.h File Reference	947
26.279tkGDCMMedicalImageProperties.h File Reference	948
26.280tkGDCMPolyDataReader.h File Reference	948
26.281tkGDCMPolyDataWriter.h File Reference	948
26.282tkGDCMTesting.h File Reference	948
26.283tkGDCMThreadedImageReader.h File Reference	949
26.284tkGDCMThreadedImageReader2.h File Reference	949
26.285tkImageColorViewer.h File Reference	949
26.286tkImageMapToColors16.h File Reference	949
26.287tkImageMapToWindowLevelColors2.h File Reference	949
26.288tkImagePlanarComponentsToComponents.h File Reference	949
26.289tkImageRGBToYBR.h File Reference	949
26.290tkImageYBRToRGB.h File Reference	950
26.291tkLookupTable16.h File Reference	950
26.292tkRTStructSetProperties.h File Reference	950
27 Example Documentation	951
27.1 AWTMedical3.java	951
27.2 BasicAnonymizer.cs	955
27.3 CastConvertPhilips.py	957
27.4 ChangeSequenceUltrasound.cxx	959
27.5 CheckBigEndianBug.cxx	960
27.6 ClinicalTrialAnnotate.cxx	962
27.7 ClinicalTrialIdentificationWorkflow.cs	963
27.8 CompressImage.cxx	966
27.9 CompressLossyJPEG.cs	967
27.10Convert16BitsTo8Bits.cxx	969
27.11ConvertMPL.py	969
27.12ConvertMultiFrameToSingleFrame.cxx	971
27.13ConvertNumpy.py	972
27.14ConvertPIL.py	973
27.15ConvertRGBToLuminance.cxx	974
27.16ConvertSingleBitTo8Bits.cxx	975
27.17ConvertToQImage.cxx	976
27.18CreateARGBImage.cxx	978
27.19CreateCMYKImage.cxx	979

27.20CreateJPIPDataSet.cxx	980
27.21CreateRAWStorage.py	981
27.22csa2img.cxx	983
27.23CStoreQtProgress.cxx	985
27.24DecompressImage.cs	987
27.25DecompressImage.py	989
27.26DecompressImageMultiframe.cs	990
27.27DecompressJPEGFile.cs	991
27.28DecompressPixmap.java	993
27.29DiffFile.cxx	994
27.30DiscriminateVolume.cxx	995
27.31DumbAnonymizer.py	999
27.32DumpADAC.cxx	1000
27.33DumpGEMSMovieGroup.cxx	1005
27.34DumpToSQLITE3.cxx	1011
27.35DuplicatePCDE.cxx	1013
27.36ELSCINT1WaveToText.cxx	1016
27.37EncapsulateFileInRawData.cxx	1018
27.38ExtractEncapsulatedFile.cs	1019
27.39ExtractEncryptedContent.cxx	1020
27.40ExtractIconFromFile.cxx	1021
27.41Extracting_All_Resolution.cxx	1022
27.42Fake_Image_Using_Stream_Image_Writer.cxx	1029
27.43FindAllPatientName.py	1032
27.44FixBrokenJ2K.cxx	1033
27.45FixCommaBug.py	1034
27.46FixJAIBugJPEGLS.cxx	1035
27.47gdcmmorthoplanes.cxx	1039
27.48gdcmmreslice.cxx	1044
27.49gdcmmrtionplan.cxx	1046
27.50gdcmmrtplan.cxx	1051
27.51gdcmmscene.cxx	1055
27.52gdcmmtexture.cxx	1057
27.53gdcmmvolume.cxx	1059
27.54GenAllIVR.cxx	1060
27.55GenerateDICOMDIR.cs	1062
27.56GenerateRTSTRUCT.cxx	1063

27.57GenerateStandardSOPClasses.cxx	1065
27.58GenFakeIdentifyFile.cxx	1066
27.59GenFakeImage.cxx	1069
27.60GenLongSeqs.cxx	1071
27.61GenSeqs.cxx	1072
27.62GetArray.cs	1073
27.63GetJPEGSamplePrecision.cxx	1075
27.64GetPortionCSAHeader.py	1076
27.65GetSequenceUltrasound.cxx	1077
27.66GetSubSequenceData.cxx	1079
27.67headsq2dcm.py	1082
27.68HelloActiviz.cs	1083
27.69HelloActiviz2.cs	1084
27.70HelloActiviz3.cs	1086
27.71HelloActiviz4.cs	1086
27.72HelloActiviz5.cs	1087
27.73HelloSimple.java	1089
27.74HelloVizWorld.cxx	1089
27.75HelloVTKWorld.cs	1090
27.76HelloVTKWorld.java	1091
27.77HelloVTKWorld2.cs	1092
27.78HelloWorld.cxx	1093
27.79HelloWorld.py	1094
27.80iU22tomultisc.cxx	1095
27.81LargeVRDSExplicit.cxx	1097
27.82MagnifyFile.cxx	1099
27.83ManipulateFile.cs	1100
27.84ManipulateFile.py	1101
27.85ManipulateSequence.py	1102
27.86MergeFile.py	1103
27.87MergeTwoFiles.cxx	1104
27.88MetImageMD5Activiz.cs	1105
27.89MIPViewer.java	1107
27.90MPRViewer.java	1109
27.91MPRViewer2.java	1111
27.92MrProtocol.cxx	1116
27.93NewSequence.cs	1123

27.94NewSequence.py	1124
27.95offscreenimage.cxx	1125
27.96PatchFile.cxx	1126
27.97PhilipsPrivateRescaleInterceptSlope.py	1127
27.98PlaySound.py	1128
27.99pmsct_rgb1.cxx	1130
27.10PrivateDict.py	1133
27.10PublicDict.cxx	1134
27.10ReadAndDumpDICOMDIR.cxx	1135
27.10ReadAndDumpDICOMDIR.py	1138
27.10ReadAndPrintAttributes.cxx	1141
27.10ReadExplicitLengthSQIVR.cxx	1142
27.10ReadFiles.java	1143
27.10ReadGEMSSDO.cxx	1144
27.10ReadMultiTimesException.cxx	1147
27.10ReadSeriesIntoVTK.java	1147
27.11ReadUTF8QtDir.cxx	1149
27.11RefCounting.cs	1150
27.11ReformatFile.cs	1151
27.11RemovePrivateTags.py	1152
27.11RescaleImage.cs	1153
27.11Reslicesphere.cxx	1154
27.11ReWriteSCAsMR.py	1162
27.11Re2img.cxx	1163
27.11structapp.cxx	1166
27.11ScanDirectory.cs	1167
27.12ScanDirectory.java	1168
27.12ScanDirectory.py	1171
27.12SendFileSCU.cs	1172
27.12SimplePrint.cs	1173
27.12SimplePrintPatientName.cs	1174
27.12SimpleScanner.cxx	1175
27.12SortImage.cxx	1176
27.12SortImage.py	1178
27.12SortImage2.cs	1179
27.12StandardizeFiles.cs	1179
27.13StreamImageReaderTest.cxx	1181

27.131	TestByteSwap.cxx	1185
27.132	TestReader.cxx	1187
27.133	TestReader.py	1188
27.134	Threadgdc.cxx	1189
27.135	TraverseModules.cxx	1192
27.136	uid_unique.cxx	1193
27.137	VolumeSorter.cxx	1194
27.138	WriteBuffer.py	1196

Chapter 1

GDCM Documentation

This is the developers documentation.

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

<http://gdcm.sourceforge.net/2.2/gdcm-2.2.0.pdf>

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

<http://gdcm.sourceforge.net/2.2/gdcm-2.2.0-doc.tar.gz>

Author

Mathieu Malaterre

Chapter 2

gdcm2pnm

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

```
gdcm2vtk [options] file-in file-out
```

3.2 DESCRIPTION

The **gdcm2vtk** 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 DICOM output filename

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 CVS). Only use it,

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 `CMake`).

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
```

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 empty/remove/replace 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 `diff(1)` for example).

4.5.4.1 Irreversible Anonymization

In some very case, 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
```

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 contain 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

gdcmconv(1), gdcmdump(1), gdcminfo(1), openssl(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.
  --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:

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

5.6.3 Compressing to lossless JPEG

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

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

5.6.4 Compressing to lossy JPEG

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

```
$ gdcconv --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:

```
$ gdcconv --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:

```
$ gdcconv --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:

```
$ gdcconv --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:

```
$ gdcconv --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 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

5.6.11 Decompressing a Compressed DICOM

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

5.6.12 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
```

5.6.13 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
```

5.6.14 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 :

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

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

```
$ gdcmmconv --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://sourceforge.net/apps/mediawiki/gdcm/index.php?title=Color_Space_Transformations

5.8 Quality Control

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

You can use 3rd party tool to check the output of gdcmmconv 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 gdcmmconv.

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

For reference, gdcmmconv --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 %!dcmdump -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), **gdcmrw(1)**, **gdcminfo(1)**

5.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 6

gdcmdiff

6.1 SYNOPSIS

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

6.2 DESCRIPTION

The **gdcmdiff** command line program takes as input two DICOM file: 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 diff 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

```
gdcmdump [options] dcm_file
gdcmdump [options] dcm_directory
```

7.2 DESCRIPTION

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

- For Implicit Transfer Syntax gdcmdump will print ?? instead of the dictionary VR

gdcmdump 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).
-A --asn1      print encapsulated ASN1 structure >(0400,0520).

```

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.

```
$ qdcmdump 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 [424.3999994] # 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_1_1_31_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

```
$ qdcmdump --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-iq-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\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. filename 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)`**

8.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 9

Manipulate DICOM image file.

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
```

```
file-out    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 (8/16/32).
--sign %s        Pixel sign (0/1).
-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)

gdcmmim 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, gray, rgb)
* RLE      (rle)
* PNM      (pgm, pnm, ppm)
* JPEG-LS  (jls)
* JPEG 2000 (jp2, j2k, j2c, 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.

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 recompressed.

9.6.2 Convert RAW to DICOM

Recognized extension is .raw, .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 explicetly 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
```

9.6.3 Convert PGM/PNM/PPM to DICOM

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

```
$ gdcimg -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)

```
$ gdcimg --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 (case insensitive)

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

the image will be a Secondary Capture

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 1.jpg 2.jpg 3.jpg 4.jpg output.dcm
```

9.8 Warning

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

9.9 SEE ALSO

gdcmdump(1), gdcmdump(1), gdcmraw(1), convert(1), dd(1)

9.10 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

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

10.4.2 general options

<code>-h --help</code>	print this help text and exit
<code>-v --version</code>	print version information and exit
<code>-V --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
```

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 or not. 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 PDF to PDF/DICOM.

11.1 SYNOPSIS

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

11.2 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.

11.3 PARAMETERS

file-in PDF input filename

file-out DICOM output filename

11.4 options

11.4.1 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
```

11.5 Usage Example

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

To re-extract the encapsulated pdf file:

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

11.6 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.

11.7 SEE ALSO

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

11.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 12

Extract Data Element Value Field.

12.1 SYNOPSIS

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

12.2 DESCRIPTION

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

12.3 PARAMETERS

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

```
file-out    output filename
```

12.4 options

12.4.1 PARAMETERS

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

12.4.2 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.
```

12.4.3 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
```

12.6 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.

12.7 SEE ALSO

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

12.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 13

Scan a directory containing DICOM files.

13.1 SYNOPSIS

```
gdcmscanner [options] directory
```

13.2 DESCRIPTION

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

13.2.1 PARAMETERS

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

13.2.2 options

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

13.2.3 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
```

13.3 Typical usage

13.4 Simple usage

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

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

13.5 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 gdcmlData -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.

13.6 SEE ALSO

gdcmdump(1), **gdcmlraw(1)**

13.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 14

Tool to execute a DICOM Query/Retrieve operation

14.1 SYNOPSIS

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

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

14.2 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).

14.3 PARAMETERS

14.4 options

14.4.1 options

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

14.4.2 mode options

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

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

14.4.3 C-STORE options

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

14.4.4 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          0123,4567=VALUE for specifying search criteria (wildcard allowed)
               With --key, leave blank (ie, --key 10,10="") to retrieve values
```

14.4.5 C-MOVE options

```
-o --output      DICOM filename / directory
--port-scp       Port for incoming associations
--key            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
```

14.4.6 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

--queryhelp      print query help
```

14.4.7 environment variable

```
GDCM_ROOT_UID Root UID
```

14.5 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
```

14.6 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 ...
```

14.7 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
```

14.8 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 your 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).

14.9 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*"
```

14.10 Debugging

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

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 its 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).

14.11 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 be accessible to all your users.

14.12 C-STORE Warnings

When constructing a C-STORE operation, `gdcm SCU` will always use the Media Storage 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
```

14.13 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.

14.14 C-FIND IMAGE level (Composite Object Instance)

One should pay attention that `gdcm SCU` `--find` and `find SCU` 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:

```
$ find SCU --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
```

14.15 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 `gdcm SCU` to be UTF-8.

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

```
$ gdcm SCU --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.

14.16 SEE ALSO

`gdcmconv(1)`

14.17 COPYRIGHT

Copyright Insight Software Consortium

Chapter 15

Concatenate/Extract DICOM files.

15.1 SYNOPSIS

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

15.2 DESCRIPTION

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

15.3 PARAMETERS

file-in DICOM input filename

file-out DICOM output filename

15.4 options

15.4.1 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.
```

15.4.2 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
```

15.4.3 environment variable

GDCM_ROOT_UID Root UID

15.5 Typical usage

15.5.1 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)  
\&...
```

15.6 SEE ALSO

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

15.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 16

Simple DICOM viewer.

16.1 SYNOPSIS

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

16.2 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 5.8, but only with VTK 5.2 (or above) can only play with the widgets (as described below).

16.3 PARAMETERS

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

16.4 options

16.4.1 options

```
--force-rescale    force rescale (advanced users)
--force-spacing    force spacing (advanced users)
-r --recursive     Recursively descend directory
```

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
```

16.5 Typical usage

16.6 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.

16.7 Wiki Link

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

16.8 SEE ALSO

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

16.9 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 17

Todo List

Class gdcm::CSAHeader

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

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 18

Deprecated List

Member `gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, bool inMove=false)`

Member `gdcm::DataElement::GetSequenceOfItems () const`

Replaced by `DataElement::GetValueAsSQ()` as of GDCM 2.2.

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 19

Bug List

Class gdcM::DICOMDIRGenerator

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

Class gdcM::IPPSorter

There currently a couple of bug in this implementation:

Chapter 20

Namespace Index

20.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gdcm	103
gdcm::network	124
gdcm::SegmentHelper	129
gdcm::terminal	
Class for Terminal Allow one to print in color in a shell	129
itk	131

Chapter 21

Class Index

21.1 Class Hierarchy

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

gdcn::network::AbstractSyntax	144
gdcn::network::ApplicationContext	153
gdcn::ApplicationEntity	154
gdcn::network::ARTIMTimer	159
gdcn::ASN1	160
gdcn::network::AsynchronousOperationsWindowSub	161
gdcn::Attribute< Group, Element, TVR, TVM >	162
gdcn::Attribute< Group, Element, TVR, VM::VM1 >	169
gdcn::Attribute< Group, Element, TVR, VM::VM1_n >	176
gdcn::Attribute< Group, Element, TVR, VM::VM1_3 >	173
gdcn::Attribute< Group, Element, TVR, VM::VM1_8 >	175
gdcn::Attribute< Group, Element, TVR, VM::VM2_n >	181
gdcn::Attribute< Group, Element, TVR, VM::VM2_2n >	180
gdcn::Attribute< Group, Element, TVR, VM::VM3_n >	184
gdcn::Attribute< Group, Element, TVR, VM::VM3_3n >	183
gdcn::Base64	188
gdcn::network::BaseCompositeMessage	189
gdcn::network::CEchoRQ	218
gdcn::network::CEchoRSP	220
gdcn::network::CFindCancelRQ	221
gdcn::network::CFindRQ	223
gdcn::network::CFindRSP	224
gdcn::network::CMoveCancelRq	225
gdcn::network::CMoveRQ	226
gdcn::network::CMoveRSP	228
gdcn::network::CStoreRQ	260
gdcn::network::CStoreRSP	261
gdcn::network::BasePDU	191
gdcn::network::AAabortPDU	133
gdcn::network::AAssociateACPDU	135
gdcn::network::AAssociateRJPDU	138
gdcn::network::AAssociateRQPDU	139
gdcn::network::AReleaseRPPDU	156

gdcmm::network::AReleaseRQPDU	157
gdcmm::network::PDataTFPDU	520
gdcmm::SegmentHelper::BasicCodedEntry	196
gdcmm::BitmapToBitmapFilter	209
gdcmm::PixmapToPixmapFilter	544
gdcmm::ImageToImageFilter	425
gdcmm::ImageApplyLookupTable	397
gdcmm::ImageChangePhotometricInterpretation	399
gdcmm::ImageChangePlanarConfiguration	403
gdcmm::ImageChangeTransferSyntax	406
gdcmm::ImageFragmentSplitter	416
gdcmm::ByteBuffer	211
gdcmm::ByteSwap< T >	212
gdcmm::ByteSwapFilter	213
gdcmm::network::CFind	221
gdcmm::Coder	230
gdcmm::Codec	229
gdcmm::AudioCodec	186
gdcmm::ImageCodec	410
gdcmm::DeltaEncodingCodec	291
gdcmm::JPEG2000Codec	451
gdcmm::JPEGCodec	456
gdcmm::JPEG12Codec	447
gdcmm::JPEG16Codec	449
gdcmm::JPEG8Codec	454
gdcmm::JPEGLSCoec	460
gdcmm::KAKADUCoec	463
gdcmm::PNMCodec	549
gdcmm::PVRGCodec	571
gdcmm::RAWCodec	583
gdcmm::RLECodec	594
gdcmm::PDFCodec	526
gdcmm::CodeString	232
gdcmm::network::CompositeMessageFactory	239
gdcmm::CompositeNetworkFunctions	240
gdcmm::ConstCharWrapper	243
gdcmm::CryptographicMessageSyntax	245
gdcmm::CSAElement	247
gdcmm::CSAHeader	251
gdcmm::CSAHeaderDict	255
gdcmm::CSAHeaderDictEntry	257
gdcmm::CSAHeaderDictException	259
gdcmm::DataElement	265
gdcmm::CP246ExplicitDataElement	243
gdcmm::ExplicitDataElement	344
gdcmm::ExplicitImplicitDataElement	346
gdcmm::Fragment	374
gdcmm::BasicOffsetTable	199
gdcmm::ImplicitDataElement	432
gdcmm::Item	443
gdcmm::UNExplicitDataElement	786
gdcmm::UNExplicitImplicitDataElement	787
gdcmm::VR16ExplicitDataElement	810

gdcm::DataElementException	275
gdcm::DataSet	278
gdcm::CommandDataSet	236
gdcm::FileMetaInformation	357
gdcm::DataSetHelper	287
gdcm::Decoder	287
gdcm::Codec	229
gdcm::DefinedTerms	289
gdcm::Defs	289
gdcm::DICOMDIR	293
gdcm::DICOMDIRGenerator	294
gdcm::Dict	296
gdcm::DictConverter	299
gdcm::DictEntry	301
gdcm::Dicts	305
gdcm::network::DIMSE	308
gdcm::DirectionCosines	309
gdcm::Directory	311
gdcm::DirectoryHelper	314
gdcm::DummyValueGenerator	315
gdcm::Element< TVR, TVM >	318
gdcm::Element< TVR, VM::VM1_n >	322
gdcm::Element< TVR, VM::VM1_2 >	321
gdcm::Element< TVR, VM::VM2_n >	327
gdcm::Element< TVR, VM::VM2_2n >	325
gdcm::Element< TVR, VM::VM3_n >	331
gdcm::Element< TVR, VM::VM3_3n >	329
gdcm::Element< VR::AS, VM::VM5 >	332
gdcm::Element< VR::OB, VM::VM1_n >	318
gdcm::Element< VR::OB, VM::VM1 >	333
gdcm::Element< VR::OW, VM::VM1_n >	318
gdcm::Element< VR::OW, VM::VM1 >	334
gdcm::EncapsulatedDocument	336
gdcm::EncodingImplementation< VR::VRASCII >	336
gdcm::EncodingImplementation< VR::VRBINARY >	337
gdcm::EnumeratedValues	339
gdcm::Event	340
gdcm::AnyEvent	152
gdcm::AbortEvent	143
gdcm::AnonymizeEvent	145
gdcm::DataEvent	275
gdcm::DataSetEvent	284
gdcm::EndEvent	338
gdcm::ExitEvent	343
gdcm::InitializeEvent	434
gdcm::IterationEvent	446
gdcm::ModifiedEvent	490
gdcm::ProgressEvent	569
gdcm::StartEvent	650
gdcm::UserEvent	792
gdcm::NoEvent	504
gdcm::Exception	342

gdcmm::ParseException	515
gdcmm::Fiducials	348
gdcmm::FileDerivation	352
gdcmm::FileExplicitFilter	355
gdcmm::Filename	362
gdcmm::FilenameGenerator	364
gdcmm::FileSet	366
itk::GDCMImageIO2	376
gdcmm::Global	384
gdcmm::GroupDict	386
gdcmm::IconImageFilter	388
gdcmm::IconImageGenerator	390
gdcmm::ignore_char	392
gdcmm::ImageConverter	415
gdcmm::ImageHelper	418
gdcmm::network::ImplementationClassUIDSub	430
gdcmm::network::ImplementationUIDSub	430
gdcmm::network::ImplementationVersionNameSub	431
gdcmm::IOD	435
gdcmm::IODEntry	436
gdcmm::IODs	438
gdcmm::LO	465
gdcmm::Scanner::ltstr	471
gdcmm::Macro	472
gdcmm::Macros	473
gdcmm::network::MaximumLengthSub	474
gdcmm::MD5	475
gdcmm::MediaStorage	476
gdcmm::Module	491
gdcmm::ModuleEntry	493
gdcmm::NestedModuleEntries	502
gdcmm::Modules	496
gdcmm::Object	505
gdcmm::BaseRootQuery	193
gdcmm::FindPatientRootQuery	369
gdcmm::FindStudyRootQuery	372
gdcmm::MovePatientRootQuery	497
gdcmm::MoveStudyRootQuery	500
gdcmm::Bitmap	201
gdcmm::Pixmap	538
gdcmm::Image	393
gdcmm::Curve	263
gdcmm::File	349
gdcmm::FileWithName	368
gdcmm::LookupTable	467
gdcmm::SegmentedPaletteColorLookupTable	608
gdcmm::MeshPrimitive	487
gdcmm::Overlay	510
gdcmm::Segment	603
gdcmm::Subject	668
gdcmm::Anonymizer	147
gdcmm::Command	234
gdcmm::MemberCommand< T >	483

gdcmm::SimpleMemberCommand< T >	634
gdcmm::network::ULConnectionManager	779
gdcmm::Scanner	597
gdcmm::ServiceClassUser	628
gdcmm::Surface	670
gdcmm::Value	795
gdcmm::ByteValue	214
gdcmm::SequenceOfFragments	614
gdcmm::SequenceOfItems	619
gdcmm::OneShotReadBuf	508
gdcmm::Orientation	508
gdcmm::Parser	517
gdcmm::Patient	519
gdcmm::PDBelement	522
gdcmm::PDBHeader	524
gdcmm::network::PDUFactory	528
gdcmm::PersonName	529
gdcmm::PhotometricInterpretation	530
gdcmm::PixelFormat	533
gdcmm::Preamble	552
gdcmm::PresentationContext	553
gdcmm::network::PresentationContextAC	555
gdcmm::PresentationContextGenerator	556
gdcmm::network::PresentationContextRQ	558
gdcmm::network::PresentationDataValue	560
gdcmm::Printer	562
gdcmm::DictPrinter	303
gdcmm::Dumper	316
gdcmm::PrivateDict	565
gdcmm::PythonFilter	573
gdcmm::QueryBase	574
gdcmm::QueryImage	577
gdcmm::QueryPatient	578
gdcmm::QuerySeries	580
gdcmm::QueryStudy	582
gdcmm::QueryFactory	576
gdcmm::Reader	586
gdcmm::PixmapReader	541
gdcmm::ImageReader	421
gdcmm::SegmentReader	609
gdcmm::SurfaceReader	679
gdcmm::Rescaler	591
gdcmm::SerieHelper::Rule	596
gdcmm::SerieHelper	624
gdcmm::Series	627
gdcmm::SHA1	633
gdcmm::SimpleSubjectWatcher	637
gdcmm::SmartPointer< ObjectType >	639
gdcmm::SOPClassUIDToIOD	642
gdcmm::Sorter	643
gdcmm::IPPSorter	439
gdcmm::Spacing	647
gdcmm::Spectroscopy	649

gdcm::SplitMosaicFilter	649
gdcm::static_assert_test< x >	651
gdcm::STATIC_ASSERTION_FAILURE< true >	652
gdcm::StreamImageReader	652
gdcm::StreamImageWriter	657
gdcm::String< TDelimiter, TMaxLength, TPadChar >	662
gdcm::StringFilter	665
gdcm::Study	668
gdcm::SurfaceHelper	677
gdcm::SwapCode	684
gdcm::SwapperDoOp	685
gdcm::SwapperNoOp	686
gdcm::System	686
gdcm::Table	690
gdcm::TableEntry	691
gdcm::TableReader	692
gdcm::XMLDictReader	860
gdcm::XMLPrivateDictReader	861
gdcm::network::TableRow	694
gdcm::Tag	695
gdcm::PrivateTag	567
gdcm::TagPath	701
gdcm::Testing	703
gdcm::Trace	706
gdcm::TransferSyntax	708
gdcm::network::TransferSyntaxSub	712
gdcm::network::Transition	713
gdcm::Type	715
gdcm::UI	717
gdcm::UIDGenerator	717
gdcm::UIDs	719
gdcm::network::ULAction	738
gdcm::network::ULActionAA1	741
gdcm::network::ULActionAA2	742
gdcm::network::ULActionAA3	743
gdcm::network::ULActionAA4	744
gdcm::network::ULActionAA5	745
gdcm::network::ULActionAA6	746
gdcm::network::ULActionAA7	748
gdcm::network::ULActionAA8	749
gdcm::network::ULActionAE1	750
gdcm::network::ULActionAE2	751
gdcm::network::ULActionAE3	752
gdcm::network::ULActionAE4	753
gdcm::network::ULActionAE5	755
gdcm::network::ULActionAE6	756
gdcm::network::ULActionAE7	757
gdcm::network::ULActionAE8	758
gdcm::network::ULActionAR1	759
gdcm::network::ULActionAR10	760
gdcm::network::ULActionAR2	762
gdcm::network::ULActionAR3	763
gdcm::network::ULActionAR4	764

gdcmm::network::ULActionAR5	765
gdcmm::network::ULActionAR6	766
gdcmm::network::ULActionAR7	767
gdcmm::network::ULActionAR8	769
gdcmm::network::ULActionAR9	770
gdcmm::network::ULActionDT1	771
gdcmm::network::ULActionDT2	772
gdcmm::network::ULConnection	775
gdcmm::network::ULConnectionCallback	777
gdcmm::network::ULBasicCallback	773
gdcmm::network::ULWritingCallback	784
gdcmm::network::ULConnectionInfo	778
gdcmm::network::ULEvent	782
gdcmm::network::ULTransitionTable	783
gdcmm::Unpacker12Bits	789
gdcmm::Usage	790
gdcmm::network::UserInformation	793
gdcmm::Validate	794
gdcmm::ValueIO< TDE, TSwap, TType >	797
gdcmm::Version	798
gdcmm::VL	799
gdcmm::VM	801
gdcmm::VR	805
gdcmm::VRVLSIZE< 0 >	812
gdcmm::VRVLSIZE< 1 >	812
vtkGDCMImageReader	813
vtkGDCMThreadedImageReader	831
vtkGDCMImageWriter	819
vtkGDCMMedicalImageProperties	823
vtkGDCMPolyDataReader	824
vtkGDCMPolyDataWriter	826
vtkGDCMTesting	829
vtkGDCMThreadedImageReader2	833
vtkImageColorViewer	836
vtkImageMapToColors16	842
vtkImageMapToWindowLevelColors2	844
vtkImagePlanarComponentsToComponents	846
vtkImageRGBToYBR	847
vtkImageYBRToRGB	848
vtkLookupTable16	848
vtkRTStructSetProperties	850
gdcmm::Waveform	854
gdcmm::Writer	854
gdcmm::PixmapWriter	546
gdcmm::ImageWriter	427
gdcmm::SegmentWriter	612
gdcmm::SurfaceWriter	682

Chapter 22

Class Index

22.1 Class List

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

gdcmm::network::AAAbortPDU	
AAAbortPDU Table 9-26 A-ABORT PDU FIELDS	133
gdcmm::network::AAssociateACPDU	
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields	135
gdcmm::network::AAssociateRJPDU	
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS	138
gdcmm::network::AAssociateRQPDU	
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields	139
gdcmm::AbortEvent	143
gdcmm::network::AbstractSyntax	
AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS	144
gdcmm::AnonymizeEvent	
AnonymizeEvent Special type of event triggered during the Anonymization process	145
gdcmm::Anonymizer	
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:	147
gdcmm::AnyEvent	152
gdcmm::network::ApplicationContext	
ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Con- text can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)	153
gdcmm::ApplicationEntity	
ApplicationEntity	154
gdcmm::network::AReleaseRPPDU	
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields	156
gdcmm::network::AReleaseRQPDU	
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS	157
gdcmm::network::ARTIMTimer	
ARTIMTimer This file contains the code for the ARTIM timer	159
gdcmm::ASN1	
Class for ASN1	160
gdcmm::network::AsynchronousOperationsWindowSub	
AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WIND- OW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	161

gdcm::Attribute< Group, Element, TVR, TVM >	
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary	162
gdcm::Attribute< Group, Element, TVR, VM::VM1 >	169
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >	173
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >	175
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >	176
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >	180
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >	181
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >	183
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >	184
gdcm::AudioCodec	
AudioCodec	186
gdcm::Base64	
Class for Base64	188
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	189
gdcm::network::BasePDU	
BasePDU base class for PDUs	191
gdcm::BaseRootQuery	193
gdcm::SegmentHelper::BasicCodedEntry	
This structure defines a basic coded entry with all of its attributes	196
gdcm::BasicOffsetTable	
Class to represent a BasicOffsetTable	199
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)	201
gdcm::BitmapToBitmapFilter	
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image	209
gdcm::ByteBuffer	
ByteBuffer	211
gdcm::ByteSwap< T >	
ByteSwap	212
gdcm::ByteSwapFilter	
ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??	213
gdcm::ByteValue	
Class to represent binary value (array of bytes)	214
gdcm::network::CEchoRQ	
CEchoRQ this file defines the messages for the cecho action	218
gdcm::network::CEchoRSP	220
gdcm::network::CFind	221
gdcm::network::CFindCancelIRQ	221
gdcm::network::CFindRQ	223
gdcm::network::CFindRSP	224
gdcm::network::CMoveCancelRq	225
gdcm::network::CMoveRQ	
CMoveRQ this file defines the messages for the cmove action	226
gdcm::network::CMoveRSP	
CMoveRSP this file defines the messages for the cmove action	228
gdcm::Codec	
Codec class	229

gdcm::Coder	
Coder	230
gdcm::CodeString	
CodeString This is an implementation of DICOM VR: CS The ctor will properly Trim so that operator== is correct	232
gdcm::Command	
Command superclass for callback/observer methods	234
gdcm::CommandDataSet	
Class to represent a Command DataSet	236
gdcm::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)	239
gdcm::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:	240
gdcm::ConstCharWrapper	
Do not use me	243
gdcm::CP246ExplicitDataElement	
Class to read/write a DataElement as CP246Explicit Data Element	243
gdcm::CryptographicMessageSyntax	
Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities	245
gdcm::CSAElement	
Class to represent a CSA Element	247
gdcm::CSAHeader	
Class for CSAHeader	251
gdcm::CSAHeaderDict	
Class to represent a map of CSAHeaderDictEntry	255
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	257
gdcm::CSAHeaderDictException	259
gdcm::network::CStoreRQ	
CStoreRQ this file defines the messages for the cecho action	260
gdcm::network::CStoreRSP	261
gdcm::Curve	
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004	263
gdcm::DataElement	
Class to represent a Data Element either Implicit or Explicit	265
gdcm::DataElementException	275
gdcm::DataEvent	
DataEvent	275

gdcm::DataSet	
Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object	278
gdcm::DataSetEvent	
DataSetEvent Special type of event triggered during the DataSet store/move process	284
gdcm::DataSetHelper	
DataSetHelper (internal class, not intended for user level)	287
gdcm::Decoder	
Decoder	287
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	289
gdcm::Defs	
FIXME I do not like the name 'Defs'	289
gdcm::DeltaEncodingCodec	
DeltaEncodingCodec compression used by some private vendor	291
gdcm::DICOMDIR	
DICOMDIR class	293
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	294
gdcm::Dict	
Class to represent a map of DictEntry	296
gdcm::DictConverter	
Class to convert a .dic file into something else:	299
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	301
gdcm::DictPrinter	
DictPrinter class	303
gdcm::Dicts	
Class to manipulate the sum of knowledge (all the dict user load)	305
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)	308
gdcm::DirectionCosines	
Class to handle DirectionCosines	309
gdcm::Directory	
Class for manipulation directories	311
gdcm::DirectoryHelper	
.	314
gdcm::DummyValueGenerator	
Class for generating dummy value	315
gdcm::Dumper	
Codec class	316
gdcm::Element< TVR, TVM >	
Element class	318
gdcm::Element< TVR, VM::VM1_2 >	
.	321

gdcm::Element< TVR, VM::VM1_n >	322
gdcm::Element< TVR, VM::VM2_2n >	325
gdcm::Element< TVR, VM::VM2_n >	327
gdcm::Element< TVR, VM::VM3_3n >	329
gdcm::Element< TVR, VM::VM3_n >	331
gdcm::Element< VR::AS, VM::VM5 >	332
gdcm::Element< VR::OB, VM::VM1 >	333
gdcm::Element< VR::OW, VM::VM1 >	334
gdcm::EncapsulatedDocument	
EncapsulatedDocument	336
gdcm::EncodingImplementation< VR::VRASCII >	336
gdcm::EncodingImplementation< VR::VRBINARY >	337
gdcm::EndEvent	338
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:	339
gdcm::Event	
Superclass for callback/observer methods	340
gdcm::Exception	
Exception	342
gdcm::ExitEvent	343
gdcm::ExplicitDataElement	
Class to read/write a DataElement as Explicit Data Element	344
gdcm::ExplicitImplicitDataElement	
Class to read/write a DataElement as ExplicitImplicit Data Element	346
gdcm::Fiducials	
Fiducials	348
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	349
gdcm::FileDerivation	
FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence	352
gdcm::FileExplicitFilter	
FileExplicitFilter class After changing a file from Implicit to Explicit representation (see ImageChange-TransferSyntax) 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	355
gdcm::FileMetaInformation	
Class to represent a File Meta Information	357
gdcm::Filename	
Class to manipulate file name's	362
gdcm::FilenameGenerator	
FilenameGenerator	364
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	366
gdcm::FileWithName	
FileWithName	368
gdcm::FindPatientRootQuery	
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root	369

gdcm::FindStudyRootQuery	
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root	372
gdcm::Fragment	
Class to represent a Fragment	374
itk::GDCMImageIO2	
ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the gdcm library (currently gdcm 2.0 is used):	376
gdcm::Global	
Global	384
gdcm::GroupDict	
Class to represent the mapping from group number to its abbreviation and name	386
gdcm::IconImageFilter	
IconImageFilter This filter will extract icons from a gdcm::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	388
gdcm::IconImageGenerator	
IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of gdcm::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:	390
gdcm::ignore_char	392
gdcm::Image	
Image	393
gdcm::ImageApplyLookupTable	
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image	397
gdcm::ImageChangePhotometricInterpretation	
ImageChangePhotometricInterpretation class Class to change the Photometric Interpretation of an input DICOM	399
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	403
gdcm::ImageChangeTransferSyntax	
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM	406
gdcm::ImageCodec	
ImageCodec	410
gdcm::ImageConverter	
Image Converter	415
gdcm::ImageFragmentSplitter	
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments	416
gdcm::ImageHelper	
ImageHelper (internal class, not intended for user level)	418
gdcm::ImageReader	
ImageReader	421
gdcm::ImageToImageFilter	
ImageToImageFilter class Super class for all filter taking an image and producing an output image	425
gdcm::ImageWriter	
ImageWriter	427
gdcm::network::ImplementationClassUIDSub	
ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELD (A-ASSOCIATE-RQ)	430
gdcm::network::ImplementationUIDSub	
ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)	430

gdcmm::network::ImplementationVersionNameSub	
ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	431
gdcmm::ImplicitDataElement	
Class to represent an <i>Implicit VR</i> Data Element	432
gdcmm::InitializeEvent	434
gdcmm::IOD	
Class for representing a IOD	435
gdcmm::IODEntry	
Class for representing a IODEntry	436
gdcmm::IODs	
Class for representing a IODs	438
gdcmm::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	439
gdcmm::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	443
gdcmm::IterationEvent	446
gdcmm::JPEG12Codec	
Class to do JPEG 12bits (lossy & lossless)	447
gdcmm::JPEG16Codec	
Class to do JPEG 16bits (lossless)	449
gdcmm::JPEG2000Codec	
Class to do JPEG 2000	451
gdcmm::JPEG8Codec	
Class to do JPEG 8bits (lossy & lossless)	454
gdcmm::JPEGCodec	
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: gdcmm::JPEG8Codec, gdcmm::JPEG12Codec & gdcmm::JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case	456
gdcmm::JPEGLSCodec	
JPEG-LS	460
gdcmm::KAKADUCodec	
KAKADUCodec	463
gdcmm::LO	
LO	465
gdcmm::LookupTable	
LookupTable class	467
gdcmm::Scanner::ltstr	471
gdcmm::Macro	
Class for representing a Macro	472
gdcmm::Macros	
Class for representing a Modules	473
gdcmm::network::MaximumLengthSub	
MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	474
gdcmm::MD5	
Class for MD5	475

gdcm::MediaStorage	
MediaStorage	476
gdcm::MemberCommand< T >	
Command subclass that calls a pointer to a member function	483
gdcm::MeshPrimitive	
This class defines surface mesh primitives. It is designed from surface mesh primitives macro	487
gdcm::ModifiedEvent	490
gdcm::Module	
Class for representing a Module	491
gdcm::ModuleEntry	
Class for representing a ModuleEntry	493
gdcm::Modules	
Class for representing a Modules	496
gdcm::MovePatientRootQuery	
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root	497
gdcm::MoveStudyRootQuery	
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root	500
gdcm::NestedModuleEntries	
Class for representing a NestedModuleEntries	502
gdcm::NoEvent	504
gdcm::Object	
Object	505
gdcm::OneShotReadBuf	508
gdcm::Orientation	
Class to handle Orientation	508
gdcm::Overlay	
Overlay class	510
gdcm::ParseException	
ParseException Standard exception handling object	515
gdcm::Parser	
Parser ala XML_Parser from expat (SAX)	517
gdcm::Patient	
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54	519
gdcm::network::PDataTFPDU	
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS	520
gdcm::PDBelement	
Class to represent a PDB Element	522
gdcm::PDBHeader	
Class for PDBHeader	524
gdcm::PDFCodec	
PDFCodec class	526
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	528
gdcm::PersonName	
PersonName class	529
gdcm::PhotometricInterpretation	
Class to represent an PhotometricInterpretation	530
gdcm::PixelFormat	
PixelFormat	533
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)	538

gdcm::PixmapReader	
PixmapReader	541
gdcm::PixmapToPixmapFilter	
PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image	544
gdcm::PixmapWriter	
PixmapWriter This class will takes two inputs:	546
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/	549
gdcm::Preamble	
DICOM Preamble (Part 10)	552
gdcm::PresentationContext	
PresentationContext	553
gdcm::network::PresentationContextAC	
PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS	555
gdcm::PresentationContextGenerator	
PresentationContextGenerator This class is responsible for generating the proper Presentation-Context 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	556
gdcm::network::PresentationContextRQ	
PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS	558
gdcm::network::PresentationDataValue	
PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS	560
gdcm::Printer	
Printer class	562
gdcm::PrivateDict	
Private Dict	565
gdcm::PrivateTag	
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)	567
gdcm::ProgressEvent	
ProgressEvent Special type of event triggered during	569
gdcm::PVRGCodec	
PVRGCodec	571
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	573
gdcm::QueryBase	
QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE	574
gdcm::QueryFactory	
QueryFactory.h	576
gdcm::QueryImage	
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE	577
gdcm::QueryPatient	
QueryPatient contains: class to construct a patient-based query for c-find and c-move	578
gdcm::QuerySeries	
QuerySeries contains: class to construct a series-based query for c-find and c-move	580
gdcm::QueryStudy	
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE	582
gdcm::RAWCodec	
RAWCodec class	583
gdcm::Reader	
Reader ala DOM (Document Object Model)	586

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	591
gdcm::RLECodec	
Class to do RLE	594
gdcm::SerieHelper::Rule	596
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	597
gdcm::Segment	
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface	603
gdcm::SegmentedPaletteColorLookupTable	
SegmentedPaletteColorLookupTable class	608
gdcm::SegmentReader	
This class defines a segment reader. It reads attributes of group 0x0062	609
gdcm::SegmentWriter	
This class defines a segment writer. It writes attributes of group 0x0062	612
gdcm::SequenceOfFragments	
Class to represent a Sequence Of Fragments	614
gdcm::SequenceOfItems	
Class to represent a Sequence Of Items (value representation : SQ)	619
gdcm::SerieHelper	624
gdcm::Series	
Series	627
gdcm::ServiceClassUser	
ServiceClassUser	628
gdcm::SHA1	
Class for SHA1	633
gdcm::SimpleMemberCommand< T >	
Command subclass that calls a pointer to a member function	634
gdcm::SimpleSubjectWatcher	
SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events	637
gdcm::SmartPointer< ObjectType >	
Class for Smart Pointer	639
gdcm::SOPClassUIDToIOD	
Class convert a class SOP Class UID into IOD	642
gdcm::Sorter	
Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction	643
gdcm::Spacing	
Class for Spacing	647
gdcm::Spectroscopy	
Spectroscopy class	649

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	649
gdcm::StartEvent	650
gdcm::static_assert_test< x >	651
gdcm::STATIC_ASSERTION_FAILURE< true >	652
gdcm::StreamImageReader	
StreamImageReader	652
gdcm::StreamImageWriter	
StreamImageReader	657
gdcm::String< TDelimiter, TMaxLength, TPadChar >	
String	662
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	665
gdcm::Study	
Study	668
gdcm::Subject	
Subject	668
gdcm::Surface	
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes	670
gdcm::SurfaceHelper	677
gdcm::SurfaceReader	
This class defines a SURFACE IE reader. It reads surface mesh module attributes	679
gdcm::SurfaceWriter	
This class defines a SURFACE IE writer. It writes surface mesh module attributes	682
gdcm::SwapCode	
SwapCode representation	684
gdcm::SwapperDoOp	685
gdcm::SwapperNoOp	686
gdcm::System	
Class to do system operation	686
gdcm::Table	
Table	690
gdcm::TableEntry	
TableEntry	691
gdcm::TableReader	
Class for representing a TableReader	692
gdcm::network::TableRow	694
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)	695
gdcm::TagPath	
Class to handle a path of tag	701
gdcm::Testing	
Class for testing	703
gdcm::Trace	
Trace	706
gdcm::TransferSyntax	
Class to manipulate Transfer Syntax	708

gdcn::network::TransferSyntaxSub	
TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS	712
gdcn::network::Transition	713
gdcn::Type	
Type	715
gdcn::Ul	717
gdcn::UIDGenerator	
Class for generating unique UID	717
gdcn::UIDs	
All known uids	719
gdcn::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	738
gdcn::network::ULActionAA1	741
gdcn::network::ULActionAA2	742
gdcn::network::ULActionAA3	743
gdcn::network::ULActionAA4	744
gdcn::network::ULActionAA5	745
gdcn::network::ULActionAA6	746
gdcn::network::ULActionAA7	748
gdcn::network::ULActionAA8	749
gdcn::network::ULActionAE1	750
gdcn::network::ULActionAE2	751
gdcn::network::ULActionAE3	752
gdcn::network::ULActionAE4	753
gdcn::network::ULActionAE5	755
gdcn::network::ULActionAE6	756
gdcn::network::ULActionAE7	757
gdcn::network::ULActionAE8	758
gdcn::network::ULActionAR1	759
gdcn::network::ULActionAR10	760
gdcn::network::ULActionAR2	762
gdcn::network::ULActionAR3	763
gdcn::network::ULActionAR4	764
gdcn::network::ULActionAR5	765
gdcn::network::ULActionAR6	766
gdcn::network::ULActionAR7	767
gdcn::network::ULActionAR8	769
gdcn::network::ULActionAR9	770
gdcn::network::ULActionDT1	771
gdcn::network::ULActionDT2	772
gdcn::network::ULBasicCallback	773
gdcn::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	775
gdcn::network::ULConnectionCallback	777
gdcn::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	778

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)	779
gdcm::network::ULEvent	
ULEvent base class for network events	782
gdcm::network::ULTransitionTable	
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates	783
gdcm::network::ULWritingCallback	784
gdcm::UNExplicitDataElement	
Class to read/write a DataElement as UNExplicit Data Element	786
gdcm::UNExplicitImplicitDataElement	
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs: 787	
gdcm::Unpacker12Bits	
Pack/Unpack 12 bits pixel into 16bits	789
gdcm::Usage	
Usage	790
gdcm::UserEvent	792
gdcm::network::UserInformation	
UserInformation Table 9-16 USER INFORMATION ITEM FIELDS	793
gdcm::Validate	
Validate class	794
gdcm::Value	
Class to represent the value of a Data Element	795
gdcm::ValueIO< TDE, TSwap, TType >	
Class to dispatch template calls	797
gdcm::Version	
Major/minor and build version	798
gdcm::VL	
Value Length	799
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	801
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	805
gdcm::VR16ExplicitDataElement	
Class to read/write a DataElement as Explicit Data Element	810
gdcm::VRVLSize< 0 >	812
gdcm::VRVLSize< 1 >	812
vtkGDCMImageReader	813
vtkGDCMImageWriter	819
vtkGDCMMedicalImageProperties	823
vtkGDCMPolyDataReader	824
vtkGDCMPolyDataWriter	826
vtkGDCMTesting	829
vtkGDCMThreadedImageReader	831
vtkGDCMThreadedImageReader2	833
vtkImageColorViewer	836
vtkImageMapToColors16	842
vtkImageMapToWindowLevelColors2	844
vtkImagePlanarComponentsToComponents	846
vtkImageRGBToYBR	847

vtkImageYBRToRGB	848
vtkLookupTable16	848
vtkRTStructSetProperties	850
gdcm::Waveform	
Waveform class	854
gdcm::Writer	
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only	854
gdcm::XMLDictReader	
Class for representing a XMLDictReader	860
gdcm::XMLPrivateDictReader	
Class for representing a XMLPrivateDictReader	861

Chapter 23

File Index

23.1 File List

Here is a list of all files with brief descriptions:

gdcm2pnm.man	865
gdcm2vtk.man	865
gdcmAAabortPDU.h	865
gdcmAAAssociateACPDU.h	865
gdcmAAAssociateRJPDU.h	866
gdcmAAAssociateRQPDU.h	866
gdcmAbstractSyntax.h	866
gdcmanon.man	866
gdcmAnonymizeEvent.h	866
gdcmAnonymizer.h	867
gdcmApplicationContext.h	867
gdcmApplicationEntity.h	867
gdcmAReleaseRPPDU.h	867
gdcmAReleaseRQPDU.h	868
gdcmARTIMTimer.h	868
gdcmASN1.h	868
gdcmAsynchronousOperationsWindowSub.h	869
gdcmAttribute.h	869
gdcmAudioCodec.h	869
gdcmBase64.h	870
gdcmBaseCompositeMessage.h	870
gdcmBasePDU.h	870
gdcmBaseRootQuery.h	870
gdcmBasicOffsetTable.h	871
gdcmBitmap.h	871
gdcmBitmapToBitmapFilter.h	872
gdcmByteBuffer.h	872
gdcmByteSwap.h	872
gdcmByteSwapFilter.h	872
gdcmByteValue.h	873
gdcmCEchoMessages.h	873
gdcmCFindMessages.h	873
gdcmCMoveMessages.h	873
gdcmCodec.h	874

gdcmCoder.h	874
gdcmCodeString.h	874
gdcmCommand.h	875
gdcmCommandDataSet.h	875
gdcmCompositeMessageFactory.h	875
gdcmCompositeNetworkFunctions.h	876
gdcmConstCharWrapper.h	876
gdcmconv.man	876
gdcmCP246ExplicitDataElement.h	876
gdcmCryptographicMessageSyntax.h	876
gdcmCSAElement.h	877
gdcmCSAHeader.h	877
gdcmCSAHeaderDict.h	877
gdcmCSAHeaderDictEntry.h	878
gdcmCStoreMessages.h	878
gdcmCurve.h	878
gdcmDataElement.h	879
gdcmDataEvent.h	879
gdcmDataSet.h	879
gdcmDataSetEvent.h	880
gdcmDataSetHelper.h	880
gdcmDecoder.h	880
gdcmDefinedTerms.h	880
gdcmDeflateStream.h	881
gdcmDefs.h	881
gdcmDeltaEncodingCodec.h	881
gdcmDICOMDIR.h	881
gdcmDICOMDIRGenerator.h	882
gdcmDict.h	882
gdcmDictConverter.h	882
gdcmDictEntry.h	883
gdcmDictPrinter.h	883
gdcmDicts.h	883
gdcmdiff.man	884
gdcmDIMSE.h	884
gdcmDirectionCosines.h	884
gdcmDirectory.h	884
gdcmDirectoryHelper.h	885
gdcmDummyValueGenerator.h	885
gdcmdump.man	885
gdcmDumper.h	885
gdcmElement.h	885
gdcmEncapsulatedDocument.h	886
gdcmEnumeratedValues.h	886
gdcmEvent.h	887
gdcmException.h	888
gdcmExplicitDataElement.h	888
gdcmExplicitImplicitDataElement.h	888
gdcmFiducials.h	888
gdcmFile.h	889
gdcmFileDerivation.h	889
gdcmFileExplicitFilter.h	889
gdcmFileMetaInformation.h	890
gdcmFilename.h	890

gdcmFilenameGenerator.h	890
gdcmFileSet.h	890
gdcmFindPatientRootQuery.h	891
gdcmFindStudyRootQuery.h	891
gdcmFragment.h	891
gdcmgendir.man	892
gdcmGlobal.h	892
gdcmGroupDict.h	892
gdcmIconImage.h	892
gdcmIconImageFilter.h	893
gdcmIconImageGenerator.h	893
gdcmImage.h	893
gdcmImageApplyLookupTable.h	894
gdcmImageChangePhotometricInterpretation.h	894
gdcmImageChangePlanarConfiguration.h	894
gdcmImageChangeTransferSyntax.h	894
gdcmImageCodec.h	895
gdcmImageConverter.h	895
gdcmImageFragmentSplitter.h	895
gdcmImageHelper.h	895
gdcmImageReader.h	896
gdcmImageToImageFilter.h	896
gdcmImageWriter.h	896
gdcmimg.man	896
gdcmImplementationClassUIDSub.h	896
gdcmImplementationUIDSub.h	897
gdcmImplementationVersionNameSub.h	897
gdcmImplicitDataElement.h	897
gdcminfo.man	898
gdcmIOD.h	898
gdcmIODEntry.h	898
gdcmIODs.h	898
gdcmIPPSorter.h	899
gdcmItem.h	899
gdcmJPEG12Codec.h	899
gdcmJPEG16Codec.h	900
gdcmJPEG2000Codec.h	900
gdcmJPEG8Codec.h	900
gdcmJPEGCodec.h	900
gdcmJPEGLSCCodec.h	901
gdcmKAKADUCodec.h	901
gdcmLegacyMacro.h	901
gdcmLO.h	901
gdcmLookupTable.h	902
gdcmMacro.h	902
gdcmMacroEntry.h	902
gdcmMacros.h	902
gdcmMaximumLengthSub.h	903
gdcmMD5.h	903
gdcmMediaStorage.h	903
gdcmMeshPrimitive.h	904
gdcmModule.h	904
gdcmModuleEntry.h	904
gdcmModules.h	905

gdcmMovePatientRootQuery.h	905
gdcmMoveStudyRootQuery.h	905
gdcmNestedModuleEntries.h	906
gdcmNetworkEvents.h	906
gdcmNetworkStateID.h	907
gdcmObject.h	907
gdcmOrientation.h	908
gdcmOverlay.h	908
gdcmParseException.h	908
gdcmParser.h	909
gdcmPatient.h	909
gdcmPDataTFPDU.h	909
gdcmPDBelement.h	909
gdcmPDBHeader.h	910
gdcmpdf.man	910
gdcmPDFCodec.h	910
gdcmPDUFactory.h	910
gdcmPersonName.h	911
gdcmPhotometricInterpretation.h	911
gdcmPixelFormat.h	911
gdcmPixmap.h	912
gdcmPixmapReader.h	912
gdcmPixmapToPixmapFilter.h	912
gdcmPixmapWriter.h	912
gdcmPNMCodec.h	913
gdcmPreamble.h	913
gdcmPresentationContext.h	913
gdcmPresentationContextAC.h	913
gdcmPresentationContextGenerator.h	914
gdcmPresentationContextRQ.h	914
gdcmPresentationDataValue.h	914
gdcmPrinter.h	915
gdcmPrivateTag.h	915
gdcmProgressEvent.h	915
gdcmPVRGCodec.h	915
gdcmPythonFilter.h	916
gdcmQueryBase.h	916
gdcmQueryFactory.h	916
gdcmQueryImage.h	917
gdcmQueryPatient.h	917
gdcmQuerySeries.h	917
gdcmQueryStudy.h	918
gdcmraw.man	918
gdcmRAWCodec.h	918
gdcmReader.h	918
gdcmRescaler.h	919
gdcmRLECodec.h	919
gdcmScanner.h	919
gdcmscanner.man	920
gdcmscu.man	920
gdcmSegment.h	920
gdcmSegmentedPaletteColorLookupTable.h	920
gdcmSegmentHelper.h	920
gdcmSegmentReader.h	920

gdcmSegmentWriter.h	921
gdcmSequenceOfFragments.h	921
gdcmSequenceOfItems.h	921
gdcmSerieHelper.h	921
gdcmSeries.h	922
gdcmServiceClassUser.h	922
gdcmSHA1.h	923
gdcmSimpleSubjectWatcher.h	923
gdcmSmartPointer.h	923
gdcmSOPClassUIDToIOD.h	923
gdcmSorter.h	924
gdcmSpacing.h	924
gdcmSpectroscopy.h	924
gdcmSplitMosaicFilter.h	924
gdcmStaticAssert.h	925
gdcmStreamImageReader.h	925
gdcmStreamImageWriter.h	926
gdcmString.h	926
gdcmStringFilter.h	926
gdcmStudy.h	927
gdcmSubject.h	927
gdcmSurface.h	927
gdcmSurfaceHelper.h	927
gdcmSurfaceReader.h	928
gdcmSurfaceWriter.h	928
gdcmSwapCode.h	928
gdcmSwapper.h	928
gdcmSystem.h	929
gdcmTable.h	929
gdcmTableEntry.h	929
gdcmTableReader.h	929
gdcmTag.h	930
gdcmTagPath.h	930
gdcmTagToVR.h	930
gdcm.tar.man	931
gdcmTerminal.h	931
gdcmTestDriver.h	931
gdcmTesting.h	931
gdcmTrace.h	932
gdcmTransferSyntax.h	934
gdcmTransferSyntaxSub.h	935
gdcmType.h	935
gdcmTypes.h	935
gdcmUIDGenerator.h	935
gdcmUIDs.h	936
gdcmULAction.h	936
gdcmULActionAA.h	936
gdcmULActionAE.h	937
gdcmULActionAR.h	937
gdcmULActionDT.h	938
gdcmULBasicCallback.h	938
gdcmULConnection.h	938
gdcmULConnectionCallback.h	938
gdcmULConnectionInfo.h	939

gdcmULConnectionManager.h	939
gdcmULEvent.h	939
gdcmULTransitionTable.h	939
gdcmULWritingCallback.h	940
gdcmUNExplicitDataElement.h	940
gdcmUNExplicitImplicitDataElement.h	940
gdcmUnpacker12Bits.h	941
gdcmUsage.h	941
gdcmUserInformation.h	941
gdcmValidate.h	941
gdcmValue.h	942
gdcmValueIO.h	942
gdcmVersion.h	942
gdcmviewer.man	943
gdcmVL.h	943
gdcmVM.h	943
gdcmVR.h	944
gdcmVR16ExplicitDataElement.h	945
gdcmWaveform.h	945
gdcmWin32.h	945
gdcmWriter.h	946
gdcmXMLDictReader.h	946
gdcmXMLPrivateDictReader.h	946
itkGDCMImageIO2.h	946
vtkGDCMImageReader.h	947
vtkGDCMImageWriter.h	947
vtkGDCMMedicalImageProperties.h	948
vtkGDCMPolyDataReader.h	948
vtkGDCMPolyDataWriter.h	948
vtkGDCMTesting.h	948
vtkGDCMThreadedImageReader.h	949
vtkGDCMThreadedImageReader2.h	949
vtkImageColorViewer.h	949
vtkImageMapToColors16.h	949
vtkImageMapToWindowLevelColors2.h	949
vtkImagePlanarComponentsToComponents.h	949
vtkImageRGBToYBR.h	949
vtkImageYBRToRGB.h	950
vtkLookupTable16.h	950
vtkRTStructSetProperties.h	950

Chapter 24

Namespace Documentation

24.1 gdcm Namespace Reference

Namespaces

- namespace network
- namespace SegmentHelper
- namespace 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 BaseRootQuery
- 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 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 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 CryptographicMessageSyntax
 - Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.*
- class CSAChecksum
 - 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 gdcm::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
- 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 EncapsulatedDocument
EncapsulatedDocument.
- 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 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 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 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 gdcm::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 gdcm::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:
- struct ignore_char
- class Image
Image.
- 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 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: gdcm::JPEG8Codec, gdcm::JPEG12Codec & gdcm::JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.
- class JPEGLSCCodec
JPEG-LS.
- 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 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 Object
 - Object.*
- struct OneShotReadBuf
- 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 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)

- 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

- 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< true >
- class StreamImageReader
StreamImageReader.
- class StreamImageWriter
StreamImageReader.
- 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
- 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 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.*
- 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.*
- class VRVLSize< 0 >
- class VRVLSize< 1 >
- class Waveform
 - Waveform class.*
- 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 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 EQueryLevel {
 ePatient,
 eStudy,
 eSeries,
 eImageOrFrame }

BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root name and date: 18 oct 2010 mmr.

- enum EQueryType {
 eFind,
 eMove }
- 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, 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 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 Dicts &d)
- 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 PDBHeader &d)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- std::ostream & operator<< (std::ostream &os, const PrivateTag &val)
- 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 Directory &d)

- `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 CSAElement &val)`
- `std::ostream & operator<< (std::ostream &os, const CSAHeader &d)`
- `std::ostream & operator<< (std::ostream &os, const VL &val)`
- `std::ostream & operator<< (std::ostream &_os, const TransferSyntax &ts)`
- `std::ostream & operator<< (std::ostream &os, const FileMetaInformation &val)`
- `std::ostream & operator<< (std::ostream &_os, const VM &_val)`
- `std::ostream & operator<< (std::ostream &os, const Fragment &val)`
- `std::ostream & operator<< (std::ostream &os, const Scanner &s)`
- `std::ostream & operator<< (std::ostream &_os, const MediaStorage &ms)`
- `std::ostream & operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)`
- `std::ostream & operator<< (std::ostream &_os, const VR &val)`
- `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`
- `VRBINARY`

24.1.1 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.

24.1.2 Typedef Documentation

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

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

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

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

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

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

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

24.1.2.8 `typedef Bitmap gdcm::IconImage`

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

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

24.1.2.11 `typedef ModuleEntry gdcm::MacroEntry`

24.1.2.12 `typedef NestedModuleEntries gdcm::NestedMacroEntries`

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

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

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

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

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

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

24.1.3 Enumeration Type Documentation

24.1.3.1 `enum gdcm::CompOperators`

Enumerator:

GDCM_EQUAL

GDCM_DIFFERENT
GDCM_GREATER
GDCM_GREATEROREQUAL
GDCM_LESS
GDCM_LESSCOREQUAL

24.1.3.2 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

24.1.3.3 enum `gdcm::EQueryLevel`

BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root name and date: 18 oct 2010 mmr.

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.

Enumerator:

ePatient
eStudy
eSeries
eImageOrFrame

24.1.3.4 enum gdcm::EQueryType

Enumerator:

eFind
eMove

24.1.3.5 enum gdcm::ERootType

Enumerator:

ePatientRootType
eStudyRootType

24.1.3.6 enum gdcm::LodModeType

Enumerator:

LD_ALL
LD_NOSEQ
LD_NOSHADOW
LD_NOSHADOWSEQ

24.1.4 Function Documentation

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

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

24.1.4.2 VR::VRType gdcm::GetVRFromTag (Tag const & tag)

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

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

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

References gdcm::Version::Print().

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

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

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

References gdcm::SwapCode::GetSwapCodeString().

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

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

Generic inserter operator for Event and its subclasses.

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

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

References `gdcmm::PDBelement::NameField`, and `gdcmm::PDBelement::ValueField`.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

References `gdcm::DataElement::GetByteValue()`, `gdcm::DataElement::ValueField`, and `gdcm::DataElement::ValueLengthField`.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

References `gdcmm::MediaStorage::GetMSString()`.

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

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

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

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

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

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

References `gdcm::UI::Internal`.

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

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

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

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

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

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

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

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

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

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

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

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

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

References `gdcm::ignore_char::m_char`.

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

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

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

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

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

24.1.5 Variable Documentation

24.1.5.1 `Global gdcm::GlobalInstance` `[static]`

24.1.5.2 gdcm::VRBINARY

24.2 gdcm::network Namespace Reference

Classes

- class AAbortPDU
AAabortPDU 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 Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)
- 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 BasePDU
BasePDU base class for PDUs.
- class CEchoRQ
CEchoRQ this file defines the messages for the cecho action.
- class CEchoRSP
- class CFind
- class CFindCancelRQ
- class CFindRQ
- class CFindRSP
- 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 echo action.

- class CStoreRSP
- 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 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 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`
- 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 `ULError`

ULError base class for network events.

- class `ULTransitionTable`

ULTransitionTable The transition table of all the `ULError`s, 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

24.2.1 Enumeration Type Documentation

24.2.1.1 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*
- eUnrecognizedPDUReceived*
- eEventDoesNotExist*

24.2.1.2 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*

24.2.2 Function Documentation

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

References eSta10ReleaseCollisionAc, eSta11ReleaseCollisionRq, eSta12ReleaseCollisionAcLocal, eSta13Awaiting-Close, eSta1Idle, eSta2Open, eSta3WaitLocalAssoc, eSta4LocalAssocDone, eSta5WaitRemoteAssoc, eSta6Transfer-Ready, eSta7WaitRelease, eSta8WaitLocalRelease, eSta9ReleaseCollisionRqLocal, and eStaDoesNotExist.

24.2.3 Variable Documentation

24.2.3.1 `const int gdcmm::network::cMaxEventID = eEventDoesNotExist`

24.2.3.2 `const int gdcmm::network::cMaxStateID = 13`

24.3 gdcmm::SegmentHelper Namespace Reference

Classes

- struct BasicCodedEntry

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

24.4 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)

24.4.1 Detailed Description

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

- support VT100 compatible shell
- win32 console

24.4.2 Enumeration Type Documentation

24.4.2.1 enum gdcmm::terminal::Attribute

Enumerator:

reset
bright
dim
underline
blink
reverse
hidden

24.4.2.2 enum gdcmm::terminal::Color

Enumerator:

black
red
green
yellow
blue
magenta
cyan
white

24.4.2.3 enum gdcmm::terminal::Mode

Enumerator:

CONSOLE
VT100

24.4.3 Function Documentation

24.4.3.1 `GDCM_EXPORT std::string gdc::terminal::setattribute (Attribute att)`

24.4.3.2 `GDCM_EXPORT std::string gdc::terminal::setbgcolor (Color c)`

24.4.3.3 `GDCM_EXPORT std::string gdc::terminal::setfgcolor (Color c)`

24.4.3.4 `GDCM_EXPORT void gdc::terminal::setmode (Mode m)`

24.5 itk Namespace Reference

Classes

- class `GDCMImageIO2`

ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the gdc library (currently gdc 2.0 is used):

Chapter 25

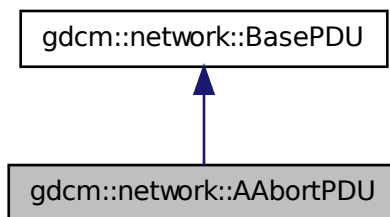
Class Documentation

25.1 gdcmm::network::AAabortPDU Class Reference

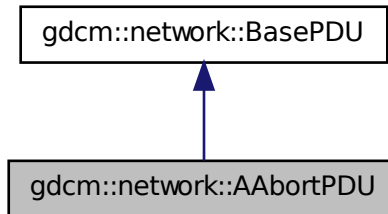
AAabortPDU Table 9-26 A-ABORT PDU FIELDS.

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

Inheritance diagram for gdcmm::network::AAabortPDU:



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)`
- `size_t Size () const`
- `const std::ostream & Write (std::ostream &os) const`

25.1.1 Detailed Description

`AAabortPDU` Table 9-26 A-ABORT PDU FIELDS.

25.1.2 Constructor & Destructor Documentation

25.1.2.1 `gdcm::network::AAabortPDU::AAabortPDU ()`

25.1.3 Member Function Documentation

25.1.3.1 `bool gdcm::network::AAabortPDU::IsLastFragment () const` `[inline], [virtual]`

Implements `gdcm::network::BasePDU`.

25.1.3.2 `void gdcm::network::AAabortPDU::Print (std::ostream & os) const` `[virtual]`

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

25.1.3.4 `size_t gdcm::network::AAbortPDU::Size () const` [virtual]

Implements `gdcm::network::BasePDU`.

25.1.3.5 `const std::ostream& gdcm::network::AAbortPDU::Write (std::ostream & os) const` [virtual]

Implements `gdcm::network::BasePDU`.

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

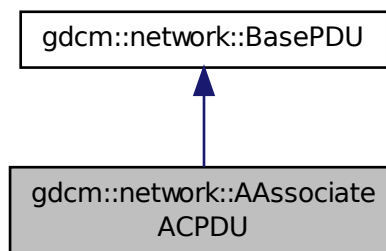
- `gdcmAAbortPDU.h`

25.2 gdcm::network::AAssociateACPDU Class Reference

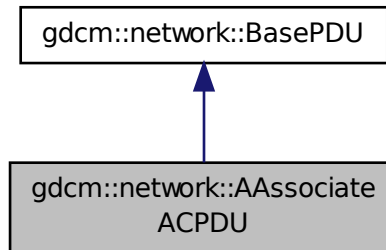
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

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

Inheritance diagram for `gdcm::network::AAssociateACPDU`:



Collaboration diagram for `gdcmm::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`

25.2.1 Detailed Description

AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

25.2.2 Member Typedef Documentation

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

25.2.3 Constructor & Destructor Documentation

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

25.2.4 Member Function Documentation

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

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

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

25.2.4.4 `const UserInformation& gdcm::network::AAssociateACPDU::GetUserInformation () const [inline]`

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

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

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

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

25.2.5 Friends And Related Function Documentation

25.2.5.1 friend class **AAssociateRQPDU** [friend]

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

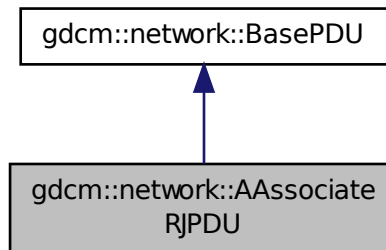
- gdcmAAssociateACPDU.h

25.3 gdcmm::network::AAssociateRJPDU Class Reference

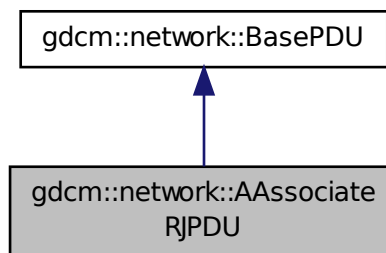
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

```
#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

25.3.1 Detailed Description

AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

25.3.2 Constructor & Destructor Documentation

25.3.2.1 gdcm::network::AAssociateRJPDU::AAssociateRJPDU ()

25.3.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

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

Implements gdcm::network::BasePDU.

25.3.3.3 std::istream& gdcm::network::AAssociateRJPDU::Read (std::istream & is) [virtual]

Implements gdcm::network::BasePDU.

25.3.3.4 size_t gdcm::network::AAssociateRJPDU::Size () const [virtual]

Implements gdcm::network::BasePDU.

25.3.3.5 const std::ostream& gdcm::network::AAssociateRJPDU::Write (std::ostream & os) const [virtual]

Implements gdcm::network::BasePDU.

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

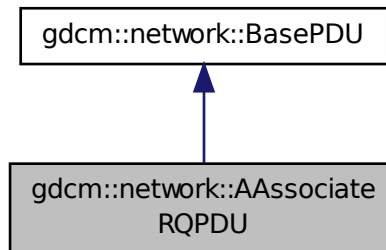
- gdcmAAssociateRJPDU.h

25.4 gdcm::network::AAssociateRQPDU Class Reference

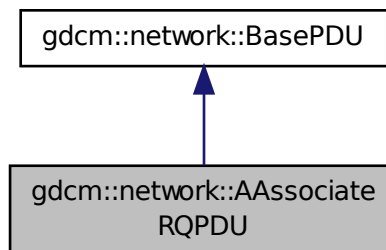
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

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

Inheritance diagram for `gdc::network::AAssociateRQPDU`:



Collaboration diagram for `gdc::network::AAssociateRQPDU`:



Public Types

- `typedef std::vector< PresentationContextRQ > PresentationContextArrayType`
- `typedef std::vector< PresentationContextRQ >::size_type SizeType`

Public Member Functions

- `AAssociateRQPDU ()`
- `AAssociateRQPDU (const AAssociateRQPDU &)`
- `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 &as) const
- const PresentationContextRQ * GetPresentationContextByID (uint8_t i) const
- PresentationContextArrayType
const & GetPresentationContexts ()
- 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.
- 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 title is a valid AE title.

25.4.1 Detailed Description

AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

25.4.2 Member Typedef Documentation

- 25.4.2.1 `typedef std::vector<PresentationContextRQ> gdcm::network::AAssociateRQPDU::PresentationContext-Array`
`Array`
`Type`
- 25.4.2.2 `typedef std::vector<PresentationContextRQ>::size_type gdcm::network::AAssociateRQPDU::SizeType`

25.4.3 Constructor & Destructor Documentation

- 25.4.3.1 `gdcm::network::AAssociateRQPDU::AAssociateRQPDU ()`
- 25.4.3.2 `gdcm::network::AAssociateRQPDU::AAssociateRQPDU (const AAssociateRQPDU &)` `[inline]`

25.4.4 Member Function Documentation

- 25.4.4.1 `void gdcm::network::AAssociateRQPDU::AddPresentationContext (PresentationContextRQ const & pc)`
- 25.4.4.2 `std::string gdcm::network::AAssociateRQPDU::GetCalledAETitle ()` `const` `[inline]`
- 25.4.4.3 `std::string gdcm::network::AAssociateRQPDU::GetCallingAETitle ()` `const` `[inline]`
- 25.4.4.4 `SizeType gdcm::network::AAssociateRQPDU::GetNumberOfPresentationContext ()` `const` `[inline]`

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

25.4.4.6 **const PresentationContextRQ*** gdcm::network::AAssociateRQPDU::GetPresentationContextByAbstractSyntax (**AbstractSyntax** const & *as*) const

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

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

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

Check whether or not the title is a valid AE title.

25.4.4.10 **bool** gdcm::network::AAssociateRQPDU::IsLastFragment () const [inline],[virtual]

Implements gdcm::network::BasePDU.

25.4.4.11 **void** gdcm::network::AAssociateRQPDU::Print (**std::ostream** & *os*) const [virtual]

This function will initialize an AAssociateACPDU from the fields in the AAssociateRQPDU structure

Implements gdcm::network::BasePDU.

25.4.4.12 **std::istream**& gdcm::network::AAssociateRQPDU::Read (**std::istream** & *is*) [virtual]

Implements gdcm::network::BasePDU.

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

Set the Called AE Title.

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

Set the Calling AE Title.

25.4.4.15 **size_t** gdcm::network::AAssociateRQPDU::Size () const [virtual]

Implements gdcm::network::BasePDU.

25.4.4.16 **const std::ostream**& gdcm::network::AAssociateRQPDU::Write (**std::ostream** & *os*) const [virtual]

Implements gdcm::network::BasePDU.

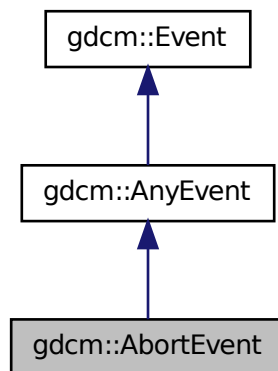
The documentation for this class was generated from the following file:

- gdcmAAssociateRQPDU.h

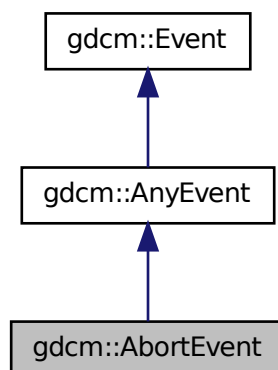
25.5 gdcm::AbortEvent Class Reference

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

Inheritance diagram for gdcm::AbortEvent:



Collaboration diagram for gdcm::AbortEvent:



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

- gdcmEvent.h

25.6 gdcmm::network::AbstractSyntax Class Reference

AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

```
#include <gdcmmAbstractSyntax.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

25.6.1 Detailed Description

AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

25.6.2 Constructor & Destructor Documentation

25.6.2.1 gdcmm::network::AbstractSyntax::AbstractSyntax ()

25.6.3 Member Function Documentation

25.6.3.1 DataElement gdcmm::network::AbstractSyntax::GetAsDataElement () const

25.6.3.2 const char* gdcmm::network::AbstractSyntax::GetName () const [inline]

25.6.3.3 bool gdcmm::network::AbstractSyntax::operator== (const AbstractSyntax & as) const [inline]

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

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

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

25.6.3.7 void gdcmm::network::AbstractSyntax::SetNameFromUID (UIDs::TSName tsname)

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

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

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

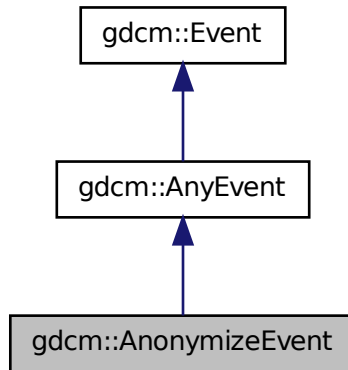
- gdcmmAbstractSyntax.h

25.7 gdcm::AnonymizeEvent Class Reference

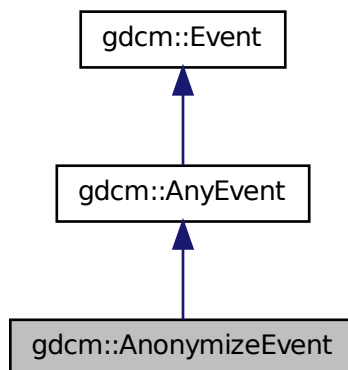
AnonymizeEvent Special type of event triggered during the Anonymization process.

```
#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 ::gdcmm::Event *e) const
- virtual const char * GetEventName () const
- Tag const & GetTag () const
- virtual ::gdcmm::Event * MakeObject () const
- void SetTag (const Tag &t)

25.7.1 Detailed Description

AnonymizeEvent Special type of event triggered during the Anonymization process.

See also

Anonymizer

25.7.2 Member Typedef Documentation

25.7.2.1 typedef AnonymizeEvent gdcmm::AnonymizeEvent::Self

25.7.2.2 typedef AnyEvent gdcmm::AnonymizeEvent::Superclass

25.7.3 Constructor & Destructor Documentation

25.7.3.1 gdcmm::AnonymizeEvent::AnonymizeEvent (Tag const & tag = 0) [inline]

25.7.3.2 virtual gdcmm::AnonymizeEvent::~~AnonymizeEvent () [inline],[virtual]

25.7.3.3 gdcmm::AnonymizeEvent::AnonymizeEvent (const Self & s) [inline]

25.7.4 Member Function Documentation

25.7.4.1 virtual bool gdcmm::AnonymizeEvent::CheckEvent (const ::gdcmm::Event * e) const [inline],[virtual]

25.7.4.2 virtual const char* gdcmm::AnonymizeEvent::GetEventName () const [inline],[virtual]

Return the StringName associated with the event.

Implements gdcmm::Event.

25.7.4.3 Tag const& gdcmm::AnonymizeEvent::GetTag () const [inline]

25.7.4.4 virtual ::gdcmm::Event* gdcmm::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 gdcmm::Event.

25.7.4.5 void gdcm::AnonymizeEvent::SetTag (const Tag & t) [inline]

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

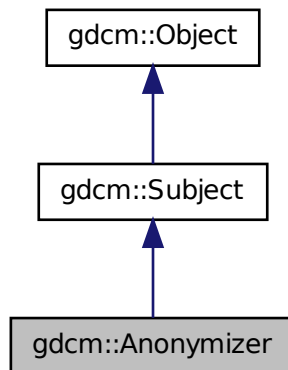
- gdcmAnonymizeEvent.h

25.8 gdcm::Anonymizer Class Reference

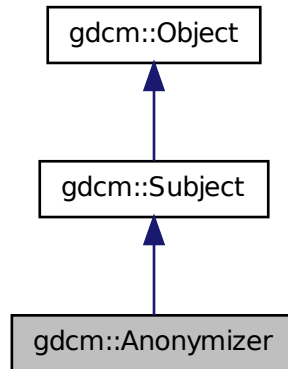
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:

```
#include <gdcmAnonymizer.h>
```

Inheritance diagram for gdcm::Anonymizer:



Collaboration diagram for gdcm::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)
remove a tag (even a SQ can be removed)
- bool RemoveGroupLength ()
Main function that loop over all elements and remove group length.
- bool RemovePrivateTags ()
Main function that loop over all elements and remove private tags.
- bool RemoveRetired ()
Main function that loop over all elements and remove retired element.
- bool Replace (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 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)

25.8.1 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 gdcm::Anonymizer class when anonymizing a FileSet. Once the gdcm::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.

25.8.2 Constructor & Destructor Documentation

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

25.8.2.2 `gdcm::Anonymizer::~~Anonymizer ()`

25.8.3 Member Function Documentation

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

25.8.3.2 `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

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

25.8.3.4 `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.

25.8.3.5 `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.

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

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

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

for wrapped language: instantiate a reference counted object

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

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

remove a tag (even a SQ can be removed)

25.8.3.11 bool gdcm::Anonymizer::RemoveGroupLength ()

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

Examples:

ClinicalTrialAnnotate.cxx.

25.8.3.12 bool gdcm::Anonymizer::RemovePrivateTags ()

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

Examples:

ClinicalTrialAnnotate.cxx.

25.8.3.13 bool gdcm::Anonymizer::RemoveRetired ()

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

25.8.3.14 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 VRASCII

Examples:

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, and EncapsulateFileInRawData.cxx.

25.8.3.15 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

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

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

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

Set/Get File.

Examples:

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, and EncapsulateFileInRawData.cxx.

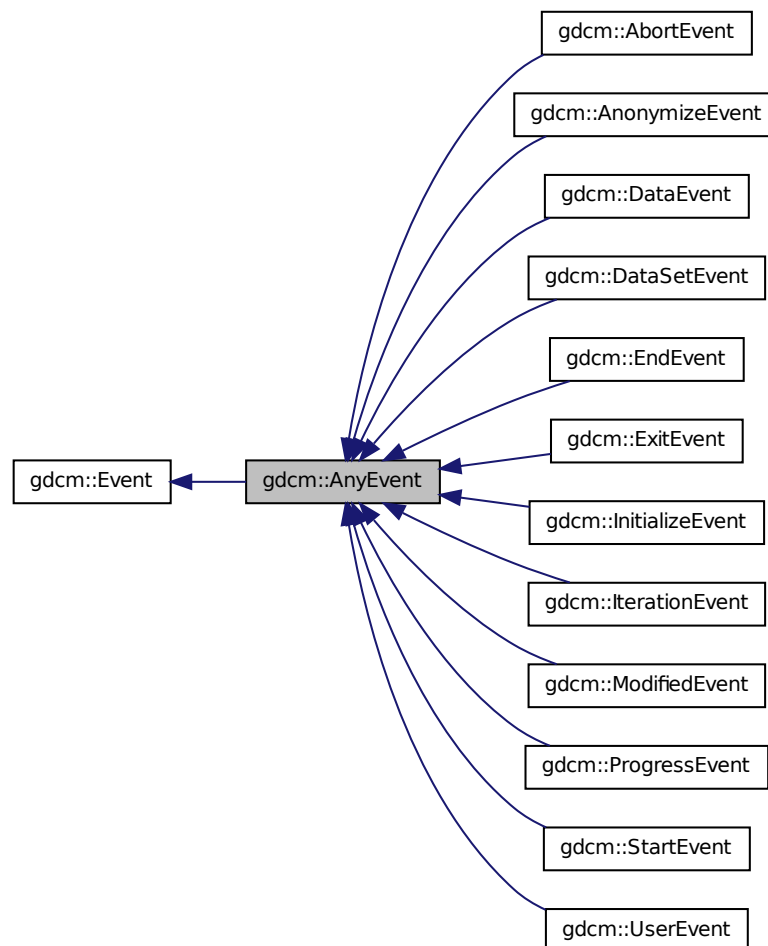
The documentation for this class was generated from the following file:

- gdcMAnonymizer.h

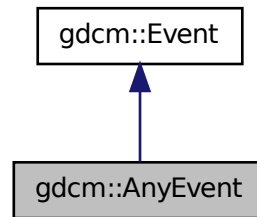
25.9 gdcM::AnyEvent Class Reference

```
#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`

25.10 gdcm::network::ApplicationContext Class Reference

ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

```
#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`

25.10.1 Detailed Description

ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

25.10.2 Constructor & Destructor Documentation

25.10.2.1 gdcm::network::ApplicationContext::ApplicationContext ()

25.10.3 Member Function Documentation

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

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

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

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

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

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

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

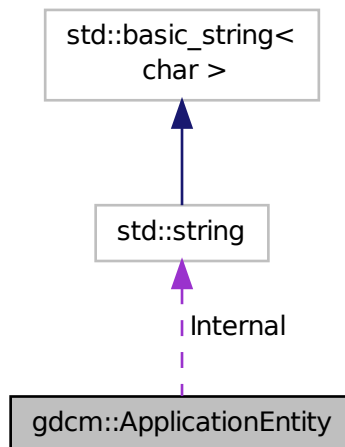
- `gdcmApplicationContext.h`

25.11 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 = ' '

25.11.1 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

25.11.2 Member Function Documentation

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

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

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

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

25.11.3 Member Data Documentation

25.11.3.1 std::string gdcmm::ApplicationEntity::Internal

25.11.3.2 const unsigned int gdcmm::ApplicationEntity::MaxLength = 16 [static]

25.11.3.3 const unsigned int gdcmm::ApplicationEntity::MaxNumberOfComponents = 1 [static]

25.11.3.4 const char gdcmm::ApplicationEntity::Padding = ' ' [static]

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

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

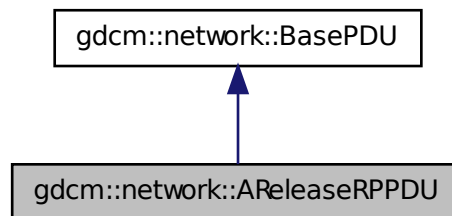
- `gdcMApplicationEntity.h`

25.12 `gdcM::network::AReleaseRPPDU` Class Reference

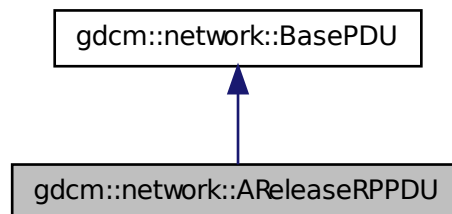
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

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

Inheritance diagram for `gdcM::network::AReleaseRPPDU`:



Collaboration diagram for `gdcM::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`

25.12.1 Detailed Description

AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

25.12.2 Constructor & Destructor Documentation

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

25.12.3 Member Function Documentation

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

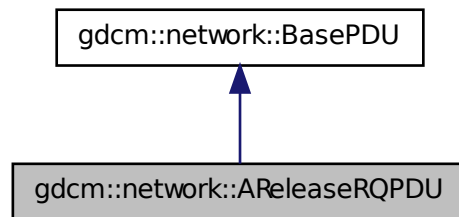
- `gdcmAReleaseRPPDU.h`

25.13 gdcm::network::AReleaseRQPDU Class Reference

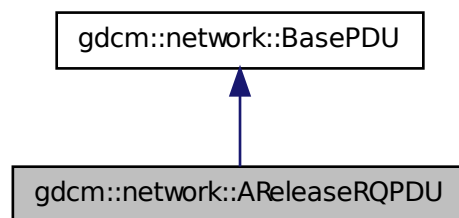
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

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

Inheritance diagram for `gdcn::network::AReleaseRQPDU`:



Collaboration diagram for `gdcn::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`

25.13.1 Detailed Description

`AReleaseRQPDU` Table 9-24 A-RELEASE-RQ PDU FIELDS.

25.13.2 Constructor & Destructor Documentation

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

25.13.3 Member Function Documentation

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

Implements `gdcm::network::BasePDU`.

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

- `gdcmAReleaseRQPDU.h`

25.14 gdcm::network::ARTIMTimer Class Reference

ARTIMTimer This file contains the code for the ARTIM timer.

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

Public Member Functions

- `ARTIMTimer ()`
- `double GetElapsedTime () const`
- `bool GetHasExpired () const`
- `double GetTimeout () const`
- `void SetTimeout (double inTimeout)`
- `void Start ()`
- `void Stop ()`

25.14.1 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.

25.14.2 Constructor & Destructor Documentation

25.14.2.1 `gdcmm::network::ARTIMTimer::ARTIMTimer ()`

25.14.3 Member Function Documentation

25.14.3.1 `double gdcmm::network::ARTIMTimer::GetElapsedTime () const`

25.14.3.2 `bool gdcmm::network::ARTIMTimer::GetHasExpired () const`

25.14.3.3 `double gdcmm::network::ARTIMTimer::GetTimeout () const`

25.14.3.4 `void gdcmm::network::ARTIMTimer::SetTimeout (double inTimeout)`

25.14.3.5 `void gdcmm::network::ARTIMTimer::Start ()`

25.14.3.6 `void gdcmm::network::ARTIMTimer::Stop ()`

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

- `gdcmmARTIMTimer.h`

25.15 gdcmm::ASN1 Class Reference

Class for ASN1.

```
#include <gdcmmASN1.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 ()

25.15.1 Detailed Description

Class for ASN1.

25.15.2 Constructor & Destructor Documentation

25.15.2.1 gdcmm::ASN1::ASN1 ()

25.15.2.2 gdcmm::ASN1::~~ASN1 ()

25.15.3 Member Function Documentation

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

25.15.3.2 static bool gdcmm::ASN1::ParseDumpFile (const char * *filename*) [static]

25.15.3.3 int gdcmm::ASN1::TestPBKDF2 () [protected]

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

- gdcmmASN1.h

25.16 gdcmm::network::AsynchronousOperationsWindowSub Class Reference

AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELD (A-ASSOCIATE-RQ)

```
#include <gdcmmAsynchronousOperationsWindowSub.h>
```

Public Member Functions

- AsynchronousOperationsWindowSub ()
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.16.1 Detailed Description

AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELD (A-ASSOCIATE-RQ)

25.16.2 Constructor & Destructor Documentation

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

25.16.3 Member Function Documentation

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

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

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

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

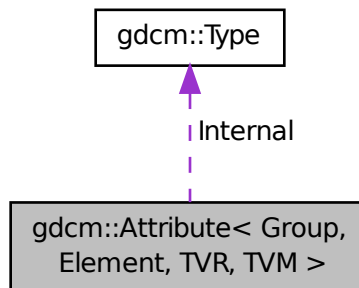
- `gdcmAsynchronousOperationsWindowSub.h`

25.17 `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)

25.17.1 Detailed Description

template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>class gdcm::Attribute< Group, Element, TVR, TVM >

Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

Typical example that compile is: Attribute<0x0008,0x9007> a = {"ORIGINAL","PRIMARY","T1","NONE"};

Examples that will NOT compile are:

Attribute<0x0018,0x1182, VR::IS, VM::VM1> fd1 = {}; // not enough parameters Attribute<0x0018,0x1182, VR::IS, VM::VM2> fd2 = {0,1,2}; // too many initializers Attribute<0x0018,0x1182, VR::IS, VM::VM3> fd3 = {0,1,2}; // VM3 is not valid Attribute<0x0018,0x1182, VR::UL, VM::VM2> fd3 = {0,1}; // UL is not valid VR

Examples:

CreateJIPIDataSet.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.-
 cxx, gdcmrtplan.cxx, GenFakeIdentifyFile.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, LargeVRDSExplicit.-
 cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndPrintAttributes.cxx, SortImage.cxx, StreamImageReaderTest.cxx,
 and VolumeSorter.cxx.

25.17.2 Member Typedef Documentation

25.17.2.1 `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`

25.17.3 Member Enumeration Documentation

25.17.3.1 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> anonymous enum`

Enumerator:

VMType

25.17.4 Member Function Documentation

25.17.4.1 `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)))`

25.17.4.2 `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)))`

25.17.4.3 `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)))`

25.17.4.4 `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::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`, `gdcm::DataElement::SetByteValue()`, `gdcm::DataElement::SetVR()`, `gdcm::VR::SQ`, `gdcm::VR::UI`, and `gdcm::VR::VRASCII`.

25.17.4.5 `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]`

25.17.4.6 `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 >::GetDictVR () [inline], [static]`

25.17.4.7 `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 >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, 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==()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::Print()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues()`.

25.17.4.8 `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]`

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::Print()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::Set()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet()`.

25.17.4.9 `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]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::operator[]()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[]()`.

25.17.4.10 `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]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

```
25.17.4.11 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]
```

References gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcmm::Attribute< Group, Element, TVR, TVM >::operator!==(), gdcmm::Attribute< Group, Element, TVR, 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==().

```
25.17.4.12 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]
```

Referenced by gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print().

```
25.17.4.13 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]
```

Referenced by gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

```
25.17.4.14 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 >::GetNumberOfValues(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

```
25.17.4.15 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(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

```
25.17.4.16 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(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

25.17.4.17 `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]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetValue()`.

25.17.4.18 `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]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetValue()`.

25.17.4.19 `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]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.17.4.20 `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()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`.

25.17.4.21 `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()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcm::ByteValue::GetPointer()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

Referenced by `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()`.

25.17.4.22 `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()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcm::ByteValue::GetPointer()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

Referenced by `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`.

```
25.17.4.23 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::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcM::DataElement::GetVR(), gdcM::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcM::VR::INVALID, gdcM::DataElement::IsEmpty(), gdcM::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcM::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), and gdcM::VR::UN.

Referenced by gdcM::Attribute< Group, Element, TVR, TVM >::Set(), gdcM::Attribute< Group, Element, TVR, VM::VM1 >::Set(), gdcM::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), and gdcM::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet().

```
25.17.4.24 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(), gdcM::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcM::DataElement::IsEmpty(), and gdcM::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

```
25.17.4.25 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]
```

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

```
25.17.4.26 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.

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

Referenced by gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), and gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues().

25.17.5 Member Data Documentation

```
25.17.5.1 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]
```

Referenced by gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute(), 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::Attribute< Group, Element, TVR, VM >::GetValue(), gdcM::Attribute< Group, Element, TVR, VM::VM1 >::GetValue(), gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue(), gdcM::Attribute< Group, Element, TVR, TVM >::GetValues(), gdcM::Attribute<

Group, Element, TVR, VM::VM1 >::GetValues(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues(), gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=(), gdcmm::Attribute< Group, Element, TVR, 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==(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==(), gdcmm::Attribute< Group, Element, TVR, TVM >::Print(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::~~Attribute().

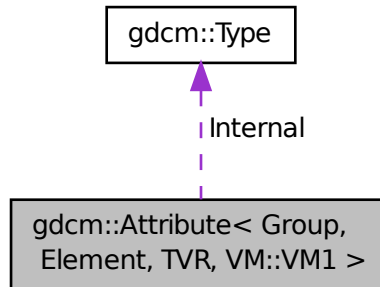
The documentation for this class was generated from the following file:

- gdcmmAttribute.h

25.18 gdcmm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference

```
#include <gdcmmAttribute.h>
```

Collaboration diagram for gdcmm::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)`

25.18.1 Member Typedef Documentation

- 25.18.1.1 `template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType`

25.18.2 Member Enumeration Documentation

- 25.18.2.1 `template<uint16_t Group, uint16_t Element, int TVR> anonymous enum`

Enumerator:

VMType

25.18.3 Member Function Documentation

25.18.3.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length ==1)`

25.18.3.2 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))`

25.18.3.3 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))`

25.18.3.4 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT ((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) VM::VM1==VM::VM1))||((VR::VRType) TVR &VR::VR_VM1)))`

25.18.3.5 `template<uint16_t Group, uint16_t Element, int TVR> DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement () const [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`, `gdcm::DataElement::SetByteValue()`, `gdcm::DataElement::SetVR()`, `gdcm::VR::SQ`, `gdcm::VR::UI`, and `gdcm::VR::VRASCII`.

25.18.3.6 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVM () [inline],[static]`

25.18.3.7 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR () [inline],[static]`

25.18.3.8 `template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues () const [inline]`

25.18.3.9 `template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag () [inline],[static]`

25.18.3.10 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue () [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.18.3.11 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue () const [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.18.3.12 `template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues () const [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.18.3.13 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM () [inline], [static]`

References gdcmm::VM::VM1.

25.18.3.14 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR () [inline], [static]`

25.18.3.15 `template<uint16_t Group, uint16_t Element, int TVR> bool gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!= (const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]`

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

25.18.3.16 `template<uint16_t Group, uint16_t Element, int TVR> bool gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator< (const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]`

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

25.18.3.17 `template<uint16_t Group, uint16_t Element, int TVR> bool gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator== (const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]`

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

25.18.3.18 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print (std::ostream & os) const [inline]`

References gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.19 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set (DataSet const & ds) [inline]`

References gdcmm::DataSet::GetDataElement(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

25.18.3.20 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (const ByteValue * bv) [inline], [protected]`

References gdcmm::ByteValue::GetLength(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcmm::ByteValue::GetPointer(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.21 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (const ByteValue * bv) [inline], [protected]`

References gdcmm::ByteValue::GetLength(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcmm::ByteValue::GetPointer(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.22 `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::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVR()`, `gdcm::VR::INVALID`, `gdcm::DataElement::IsEmpty()`, `gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`, `gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap()`, and `gdcm::VR::UN`.

25.18.3.23 `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()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::IsEmpty()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`.

25.18.3.24 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (ArrayType v) [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.18.4 Member Data Documentation

25.18.4.1 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal`

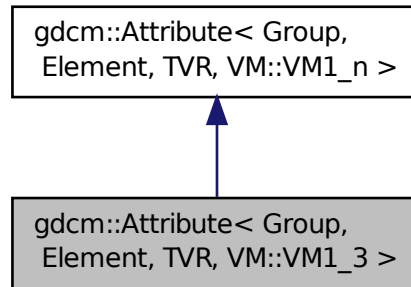
The documentation for this class was generated from the following file:

- `gdcmAttribute.h`

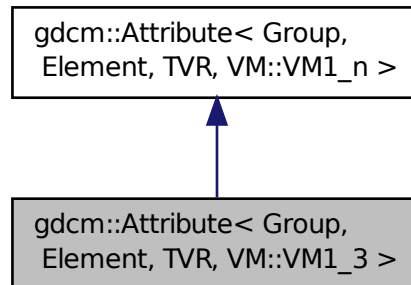
25.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

```
#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

25.19.1 Member Function Documentation

25.19.1.1 `template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM () const` [inline]

References `gdcm::VM::VM1_3`.

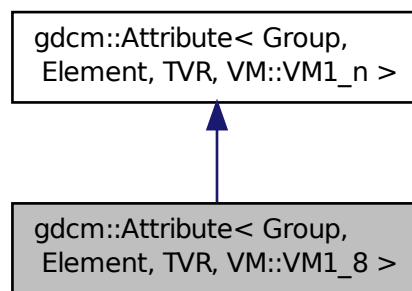
The documentation for this class was generated from the following file:

- gdcMAttribute.h

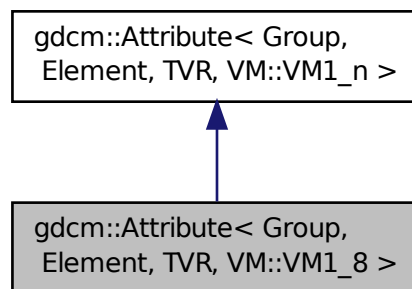
25.20 gdcM::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

```
#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

25.20.1 Member Function Documentation

25.20.1.1 `template<uint16_t Group, uint16_t Element, int TVR> VM gdcM::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM() const [inline]`

References `gdcM::VM::VM1_8`.

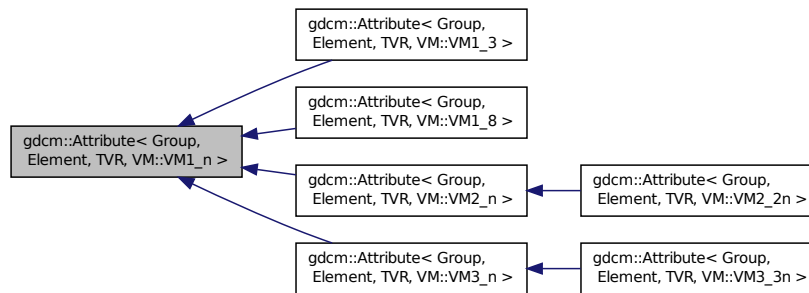
The documentation for this class was generated from the following file:

- `gdcMAttribute.h`

25.21 `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 SetFromDataElement (DataElement const &de)
- 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)

25.21.1 Member Typedef Documentation

25.21.1.1 `template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::ArrayType`

25.21.2 Constructor & Destructor Documentation

25.21.2.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute () [inline],[explicit]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.2.2 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::~~Attribute () [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3 Member Function Documentation

25.21.3.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (((VR::VRType) TVR & (VR::VRType)(TagToType< Group, Element >::VRType)))`

25.21.3.2 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT ((VM::VM1_n & (VM::VMType)(TagToType< Group, Element >::VMType)))`

25.21.3.3 `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)))`

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

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`, `gdcm::DataElement::SetByteValue()`, `gdcm::DataElement::SetVR()`, `gdcm::VR::SQ`, `gdcm::VR::UI`, and `gdcm::VR::VRASCII`.

25.21.3.5 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM () [inline], [static]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetVM()`.

25.21.3.6 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR () [inline], [static]`

25.21.3.7 `template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues () const [inline]`

25.21.3.8 `template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag () [inline], [static]`

25.21.3.9 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (unsigned int idx = 0) [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.10 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (unsigned int idx = 0) const [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.11 `template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues () const [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.12 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM () [inline], [static]`

Reimplemented in `gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >`, `gdcm::Attribute< Group, Element, TVR, VM::VM3_n >`, and `gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >`.

References `gdcm::VM::VM1_n`.

25.21.3.13 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR() [inline], [static]`

25.21.3.14 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[](unsigned int idx) [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetValue()`.

25.21.3.15 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[](unsigned int idx) const [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetValue()`.

25.21.3.16 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(std::ostream & os) const [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVM()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVR()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.17 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(const ByteValue * bv) [inline], [protected]`

References `gdcm::ByteValue::GetLength()`, `gdcm::ByteValue::GetPointer()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::SetValues()`.

25.21.3.18 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(DataElement const & de) [inline]`

References `gdcm::DataElement::GetByteValue()`, `gdcm::Tag::GetGroup()`, `gdcm::DataElement::GetTag()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVR()`, `gdcm::DataElement::IsEmpty()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`.

25.21.3.19 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues(unsigned int numel) [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::SetValues()`.

25.21.3.20 `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]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.21 `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().

```
25.21.3.22  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]
```

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

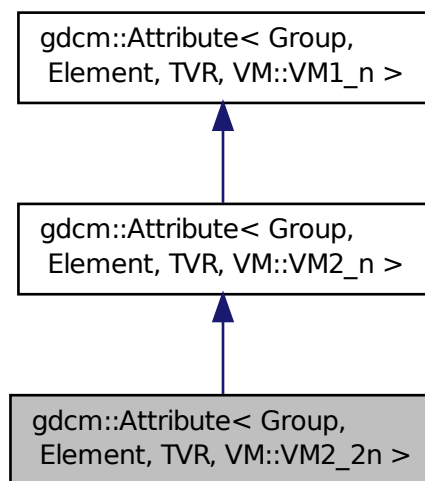
The documentation for this class was generated from the following file:

- gdcMAttribute.h

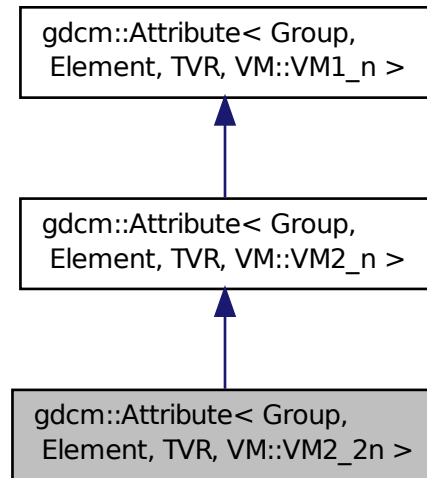
25.22 gdcM::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

```
#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

25.22.1 Member Function Documentation

25.22.1.1 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcM::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM () [inline], [static]`

Reimplemented from `gdcM::Attribute< Group, Element, TVR, VM::VM1_n >`.

References `gdcM::VM::VM2_2n`.

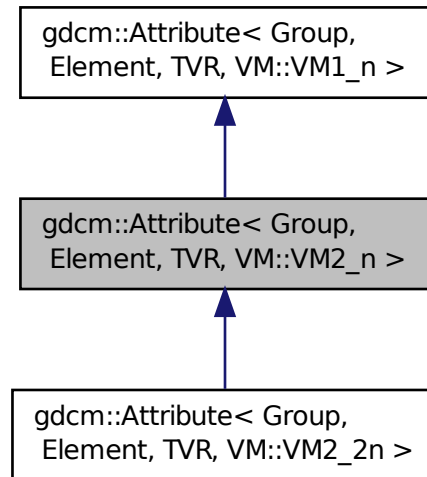
The documentation for this class was generated from the following file:

- `gdcMAttribute.h`

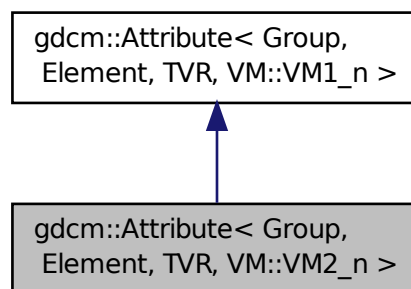
25.23 gdcM::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

```
#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

25.23.1 Member Function Documentation

25.23.1.1 `template<uint16_t Group, uint16_t Element, int TVR> VM gdcM::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM() const [inline]`

References gdcM::VM::VM2_n.

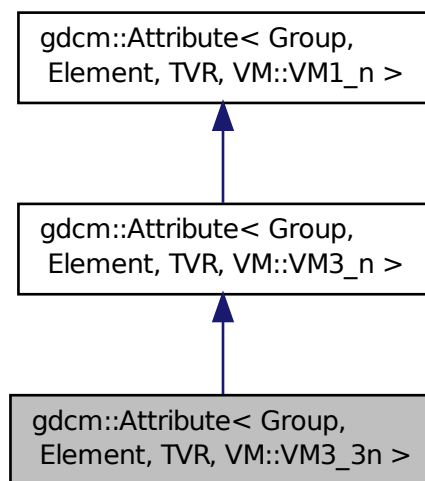
The documentation for this class was generated from the following file:

- gdcMAttribute.h

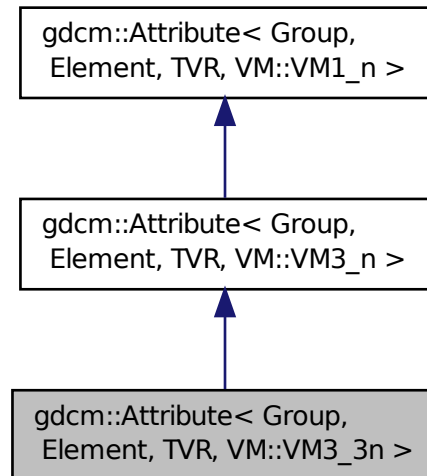
25.24 gdcM::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

```
#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 ()`

25.24.1 Member Function Documentation

25.24.1.1 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcM::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM () [inline], [static]`

Reimplemented from `gdcM::Attribute< Group, Element, TVR, VM::VM3_n >`.

References `gdcM::VM::VM3_3n`.

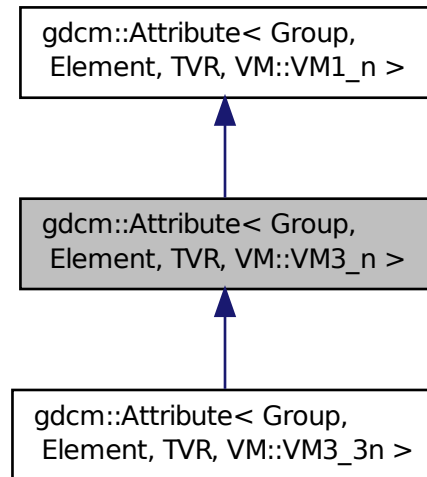
The documentation for this class was generated from the following file:

- `gdcMAttribute.h`

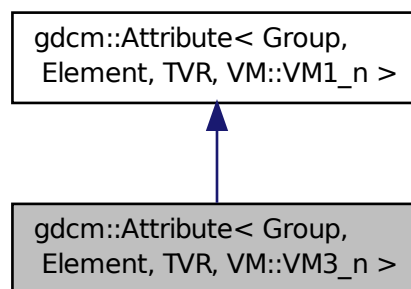
25.25 `gdcM::Attribute< Group, Element, TVR, VM::VM3_n >` Class Template Reference

```
#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

25.25.1 Member Function Documentation

25.25.1.1 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcM::Attribute< Group, Element, TVR, VM::VM3_n >::GetVM () [inline],[static]`

Reimplemented from `gdcM::Attribute< Group, Element, TVR, VM::VM1_n >`.

Reimplemented in `gdcM::Attribute< Group, Element, TVR, VM::VM3_3n >`.

References `gdcM::VM::VM3_n`.

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

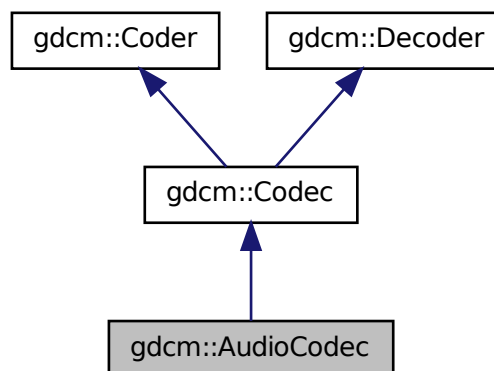
- `gdcMAttribute.h`

25.26 gdcM::AudioCodec Class Reference

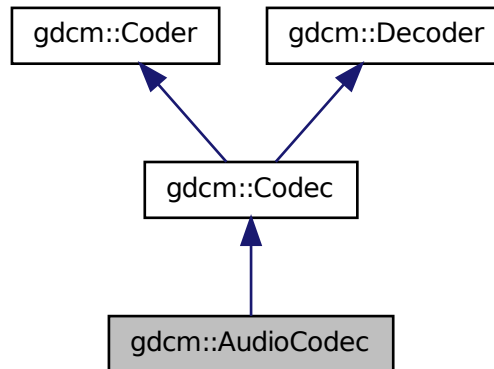
AudioCodec.

```
#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.

25.26.1 Detailed Description

`AudioCodec`.

25.26.2 Constructor & Destructor Documentation

25.26.2.1 `gdcm::AudioCodec::AudioCodec ()`

25.26.2.2 `gdcm::AudioCodec::~~AudioCodec ()`

25.26.3 Member Function Documentation

25.26.3.1 `bool gdcm::AudioCodec::CanCode (TransferSyntax const &) const` `[inline], [virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements `gdcm::Coder`.

25.26.3.2 `bool gdcm::AudioCodec::CanDecode (TransferSyntax const &) const` `[inline],[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements `gdcm::Decoder`.

25.26.3.3 `bool gdcm::AudioCodec::Decode (DataElement const & is_, DataElement & os)` `[virtual]`

Decode.

Reimplemented from `gdcm::Decoder`.

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

- `gdcmAudioCodec.h`

25.27 gdcm::Base64 Class Reference

Class for Base64.

```
#include <gdcmBase64.h>
```

Public Member Functions

- `Base64 ()`
- `~Base64 ()`

Static Public Member Functions

- `static int Decode (char *dst, int dlen, const char *src, int slen)`
Decode a base64-formatted buffer.
- `static int Encode (char *dst, int dlen, const char *src, int slen)`
Encode a buffer into base64 format.
- `static int GetDecodeLength (const char *src, int slen)`
- `static int GetEncodeLength (const char *src, int slen)`

25.27.1 Detailed Description

Class for Base64.

25.27.2 Constructor & Destructor Documentation

25.27.2.1 `gdcm::Base64::Base64 ()`

25.27.2.2 `gdcm::Base64::~~Base64 ()`

25.27.3 Member Function Documentation

25.27.3.1 `static int gdcm::Base64::Decode (char * dst, int dlen, const char * src, int slen) [static]`

Decode a base64-formatted buffer.

Parameters

<i>dst</i>	destination buffer
<i>dlen</i>	size of the buffer
<i>src</i>	source buffer
<i>slen</i>	amount of data to be decoded

Returns

0 if successful

25.27.3.2 `static int gdcm::Base64::Encode (char * dst, int dlen, const char * src, int 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 successful

25.27.3.3 `static int gdcm::Base64::GetDecodeLength (const char * src, int slen) [static]`

Call this function with *dlen = 0 to obtain the required buffer size in *dlen

25.27.3.4 `static int gdcm::Base64::GetEncodeLength (const char * src, int slen) [static]`

Call this function with dlen = 0 to obtain the required buffer size in dlen

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

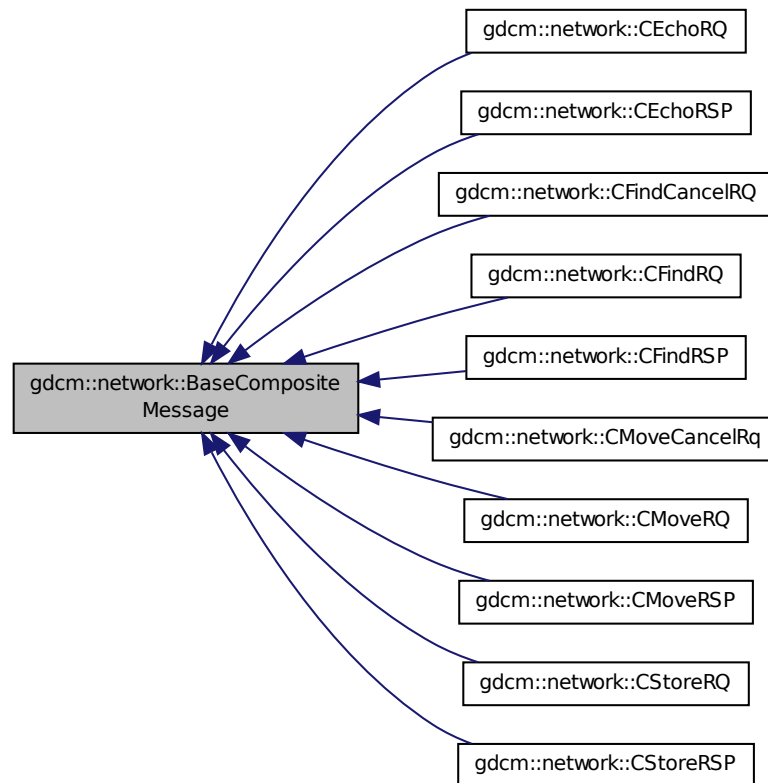
- gdcmBase64.h

25.28 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 std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0`

25.28.1 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.

25.28.2 Member Function Documentation

25.28.2.1 `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:

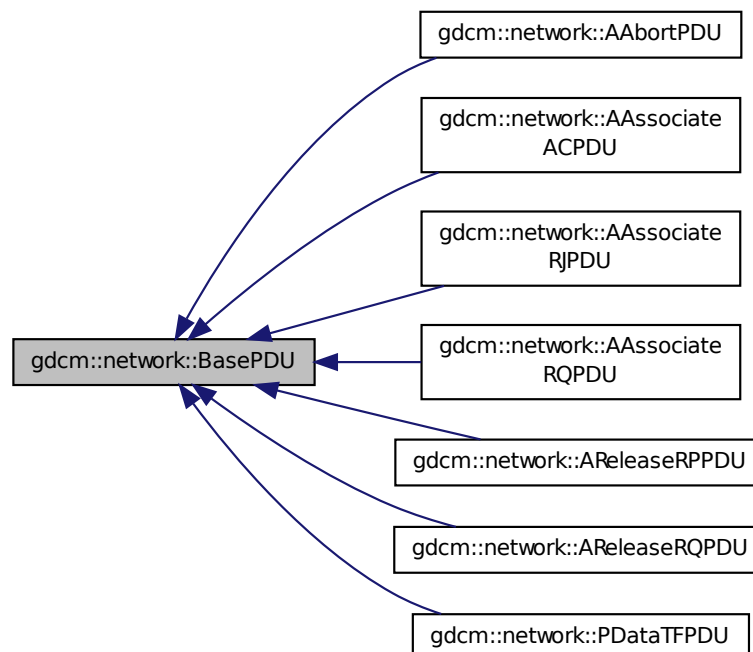
- `gdcmBaseCompositeMessage.h`

25.29 gdcm::network::BasePDU Class Reference

BasePDU base class for PDUs.

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

Inheritance diagram for `gdcm::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`

25.29.1 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

but! leave Mathieu's original classes untouched at this point (except for inheriting from this class) because those work, at least with c-echo.

25.29.2 Constructor & Destructor Documentation

25.29.2.1 `virtual gdcn::network::BasePDU::~~BasePDU () [inline], [virtual]`

25.29.3 Member Function Documentation

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

Implemented in `gdcn::network::AAssociateRQPDU`, `gdcn::network::AAssociateACPDU`, `gdcn::network::PDataTFPDU`, `gdcn::network::AAabortPDU`, `gdcn::network::AAssociateRJPDU`, `gdcn::network::AReleaseRPPDU`, and `gdcn::network::AReleaseRQPDU`.

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

Implemented in `gdcn::network::AAssociateRQPDU`, `gdcn::network::AAssociateACPDU`, `gdcn::network::PDataTFPDU`, `gdcn::network::AAabortPDU`, `gdcn::network::AReleaseRPPDU`, `gdcn::network::AReleaseRQPDU`, and `gdcn::network::AAssociateRJPDU`.

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

Implemented in `gdcn::network::AAssociateACPDU`, `gdcn::network::AAssociateRQPDU`, `gdcn::network::PDataTFPDU`, `gdcn::network::AAssociateRJPDU`, `gdcn::network::AReleaseRPPDU`, `gdcn::network::AReleaseRQPDU`, and `gdcn::network::AAabortPDU`.

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

Implemented in `gdcmm::network::AAssociateACPDU`, `gdcmm::network::AAssociateRQPDU`, `gdcmm::network::PDataTFPDU`, `gdcmm::network::AAabortPDU`, `gdcmm::network::AAssociateRJPDU`, `gdcmm::network::AReleaseRPPDU`, and `gdcmm::network::AReleaseRQPDU`.

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

Implemented in `gdcmm::network::AAssociateACPDU`, `gdcmm::network::AAssociateRQPDU`, `gdcmm::network::PDataTFPDU`, `gdcmm::network::AAssociateRJPDU`, `gdcmm::network::AReleaseRPPDU`, `gdcmm::network::AReleaseRQPDU`, and `gdcmm::network::AAabortPDU`.

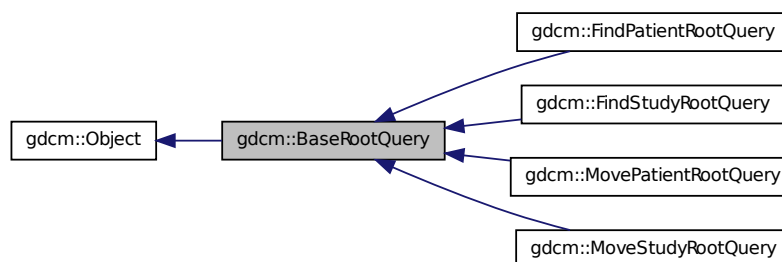
The documentation for this class was generated from the following file:

- `gdcmmBasePDU.h`

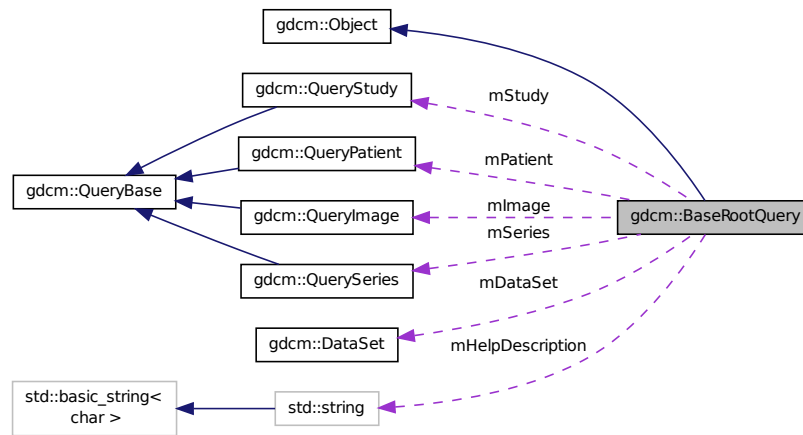
25.30 gdcmm::BaseRootQuery Class Reference

```
#include <gdcmmBaseRootQuery.h>
```

Inheritance diagram for `gdcmm::BaseRootQuery`:



Collaboration diagram for `gdcm::BaseRootQuery`:



Public Member Functions

- `virtual ~BaseRootQuery ()`
- `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 ()`
- `virtual std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)=0`
- `virtual void InitializeDataSet (const EQueryLevel &inQueryLevel)=0`
- `void SetSearchParameter (const Tag &inTag, const std::string &inValue)`
- `void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)`
- `virtual bool ValidateQuery (bool inStrict=true) const =0`
- `virtual const std::ostream & WriteHelpFile (std::ostream &os)`
- `virtual bool WriteQuery (const std::string &inFileName)`

Protected Member Functions

- `BaseRootQuery ()`
- `void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)`

Protected Attributes

- `DataSet mDataSet`
- `std::string mHelpDescription`
- `QueryImage mImage`
- `QueryPatient mPatient`
- `ERootType mRootType`
- `QuerySeries mSeries`
- `QueryStudy mStudy`

Friends

- class QueryFactory

25.30.1 Constructor & Destructor Documentation

25.30.1.1 `gdcm::BaseRootQuery::BaseRootQuery ()` [protected]

25.30.1.2 `virtual gdcm::BaseRootQuery::~~BaseRootQuery ()` [virtual]

25.30.2 Member Function Documentation

25.30.2.1 `void gdcm::BaseRootQuery::AddQueryDataSet (const DataSet & ds)`

25.30.2.2 `virtual UIDs::TSName gdcm::BaseRootQuery::GetAbstractSyntaxUID () const` [pure virtual]

Implemented in `gdcm::FindStudyRootQuery`, `gdcm::MovePatientRootQuery`, `gdcm::MoveStudyRootQuery`, and `gdcm::FindPatientRootQuery`.

25.30.2.3 `DataSet const& gdcm::BaseRootQuery::GetQueryDataSet () const`

Set/Get the internal representation of the query as a DataSet.

25.30.2.4 `DataSet& gdcm::BaseRootQuery::GetQueryDataSet ()`

25.30.2.5 `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`, and `gdcm::MoveStudyRootQuery`.

25.30.2.6 `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 dcmTk

Implemented in `gdcm::FindPatientRootQuery`, `gdcm::FindStudyRootQuery`, `gdcm::MovePatientRootQuery`, and `gdcm::MoveStudyRootQuery`.

25.30.2.7 `void gdcm::BaseRootQuery::SetSearchParameter (const Tag & inTag, const DictEntry & inDictEntry, const std::string & inValue)` [protected]

25.30.2.8 `void gdcm::BaseRootQuery::SetSearchParameter (const Tag & inTag, const std::string & inValue)`

25.30.2.9 `void gdcm::BaseRootQuery::SetSearchParameter (const std::string & inKeyword, const std::string & inValue)`

25.30.2.10 `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.

Implemented in `gdcm::FindStudyRootQuery`, `gdcm::MovePatientRootQuery`, `gdcm::MoveStudyRootQuery`, and `gdcm::FindPatientRootQuery`.

25.30.2.11 `virtual const std::ostream& gdcm::BaseRootQuery::WriteHelpFile (std::ostream & os)` `[virtual]`

25.30.2.12 `virtual bool gdcm::BaseRootQuery::WriteQuery (const std::string & inFileName)` `[virtual]`

25.30.3 Friends And Related Function Documentation

25.30.3.1 `friend class QueryFactory` `[friend]`

Reimplemented in `gdcm::FindPatientRootQuery`, `gdcm::FindStudyRootQuery`, `gdcm::MovePatientRootQuery`, and `gdcm::MoveStudyRootQuery`.

25.30.4 Member Data Documentation

25.30.4.1 `DataSet gdcm::BaseRootQuery::mDataSet` `[protected]`

25.30.4.2 `std::string gdcm::BaseRootQuery::mHelpDescription` `[protected]`

25.30.4.3 `QueryImage gdcm::BaseRootQuery::mImage` `[protected]`

25.30.4.4 `QueryPatient gdcm::BaseRootQuery::mPatient` `[protected]`

25.30.4.5 `ERootType gdcm::BaseRootQuery::mRootType` `[protected]`

25.30.4.6 `QuerySeries gdcm::BaseRootQuery::mSeries` `[protected]`

25.30.4.7 `QueryStudy gdcm::BaseRootQuery::mStudy` `[protected]`

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

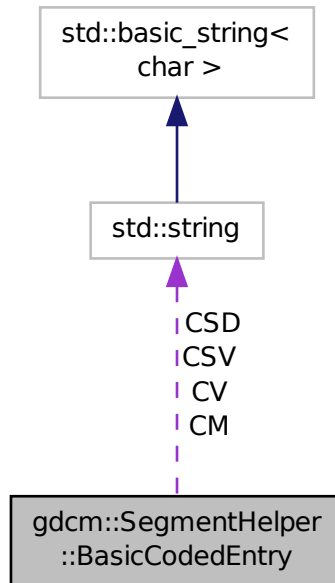
- `gdcmBaseRootQuery.h`

25.31 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

25.31.1 Detailed Description

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

See also

PS 3.3 section 8.8.

25.31.2 Constructor & Destructor Documentation

25.31.2.1 `gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ()` `[inline]`

Constructor.

25.31.2.2 `gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (const char * a_CV, const char * a_CSD, const char * a_CM)` `[inline]`

constructor which defines type 1 attributes.

25.31.2.3 `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.

25.31.3 Member Function Documentation

25.31.3.1 `bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (const bool checkOptionalAttributes = false) const`

Check if each attributes of the basic coded entry is defined.

Parameters

<i>checkOptional-Attributes</i>	Check also type 1C attributes.
---------------------------------	--------------------------------

25.31.4 Member Data Documentation

25.31.4.1 `std::string gdcm::SegmentHelper::BasicCodedEntry::CM`

Coding Scheme Version attribute.

25.31.4.2 `std::string gdcm::SegmentHelper::BasicCodedEntry::CSD`

Code Value attribute.

25.31.4.3 `std::string gdcm::SegmentHelper::BasicCodedEntry::CSV`

Coding Scheme Designator attribute.

25.31.4.4 std::string gdcm::SegmentHelper::BasicCodedEntry::CV

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

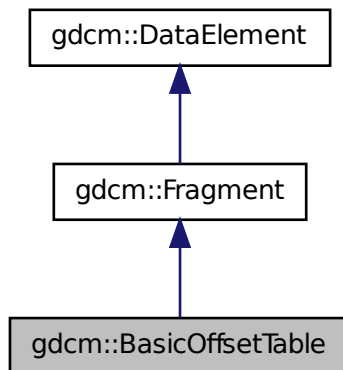
- gdcmSegmentHelper.h

25.32 gdcm::BasicOffsetTable Class Reference

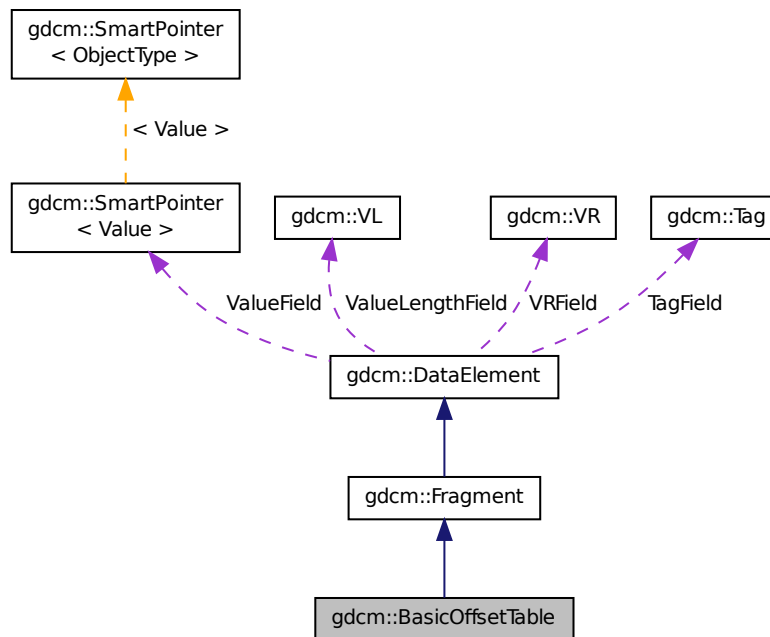
Class to represent a BasicOffsetTable.

```
#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)`

25.32.1 Detailed Description

Class to represent a `BasicOffsetTable`.

25.32.2 Constructor & Destructor Documentation

25.32.2.1 `gdcm::BasicOffsetTable::BasicOffsetTable ()` `[inline]`

25.32.3 Member Function Documentation

25.32.3.1 `template<typename TSwap> std::istream& gdcm::BasicOffsetTable::Read (std::istream & is) [inline]`

Reimplemented from `gdcm::Fragment`.

25.32.4 Friends And Related Function Documentation

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

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

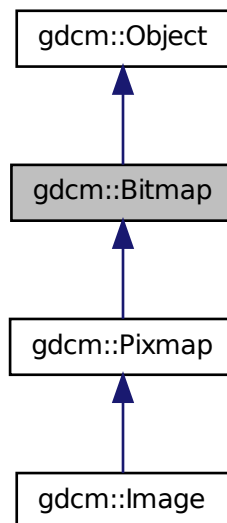
- `gdcmBasicOffsetTable.h`

25.33 gdcm::Bitmap Class Reference

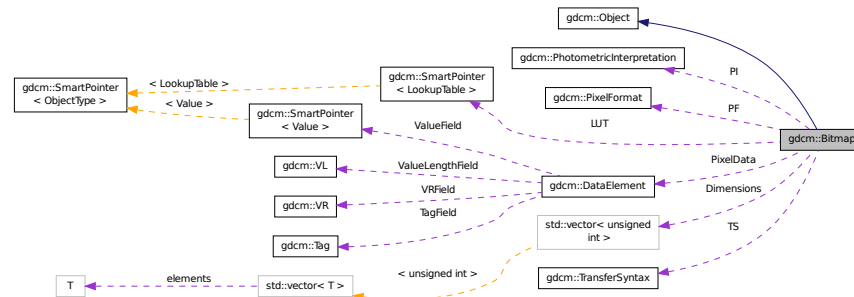
Bitmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

```
#include <gdcmBitmap.h>
```

Inheritance diagram for `gdcm::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

25.33.1 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.

25.33.2 Member Typedef Documentation

25.33.2.1 `typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr` [protected]

25.33.3 Constructor & Destructor Documentation

25.33.3.1 `gdcm::Bitmap::Bitmap ()`

25.33.3.2 `gdcm::Bitmap::~~Bitmap ()`

25.33.4 Member Function Documentation

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

Reimplemented in `gdcm::Pixmap`.

25.33.4.2 `void gdcm::Bitmap::Clear ()`

25.33.4.3 `bool gdcm::Bitmap::ComputeLossyFlag ()` [protected]

25.33.4.4 `bool gdcm::Bitmap::GetBuffer (char * buffer) const`

Acces the raw data.

Examples:

ConvertToQImage.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.33.4.5 `bool gdcm::Bitmap::GetBuffer2 (std::ostream & os) const` [protected]

25.33.4.6 `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, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.33.4.7 unsigned int gdcm::Bitmap::GetColumns () const [inline]

25.33.4.8 const DataElement& gdcm::Bitmap::GetDataElement () const [inline]

Examples:

ExtractIconFromFile.cxx.

25.33.4.9 DataElement& gdcm::Bitmap::GetDataElement () [inline]

25.33.4.10 unsigned int gdcm::Bitmap::GetDimension (unsigned int *idx*) const

25.33.4.11 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.

25.33.4.12 const LookupTable& gdcm::Bitmap::GetLUT () const [inline]

Examples:

ExtractIconFromFile.cxx.

25.33.4.13 LookupTable& gdcm::Bitmap::GetLUT () [inline]

25.33.4.14 bool gdcm::Bitmap::GetNeedByteSwap () const [inline]

25.33.4.15 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.

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

return the photometric interpretation

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, and HelloVizWorld.cxx.

25.33.4.17 `const PixelFormat& gdcm::Bitmap::GetPixelFormat () const` `[inline]`

Get/Set PixelFormat.

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAI BugJPEGLS.cxx, GenFakeImage.cxx, GetJPEGSample-Precision.cxx, and threadgdcm.cxx.

25.33.4.18 `PixelFormat& gdcm::Bitmap::GetPixelFormat ()` `[inline]`

25.33.4.19 `unsigned int gdcm::Bitmap::GetPlanarConfiguration () const`

return the planar configuration

25.33.4.20 `unsigned int gdcm::Bitmap::GetRows () const` `[inline]`

25.33.4.21 `const TransferSyntax& gdcm::Bitmap::GetTransferSyntax () const` `[inline]`

Examples:

ExtractIconFromFile.cxx.

25.33.4.22 `bool gdcm::Bitmap::IsEmpty () const` `[inline]`

25.33.4.23 `bool gdcm::Bitmap::IsLossy () const`

Return whether or not the image was compressed using a lossy compressor or not.

25.33.4.24 `bool gdcm::Bitmap::IsTransferSyntaxCompatible (TransferSyntax const & ts) const`

25.33.4.25 `void gdcm::Bitmap::Print (std::ostream &) const` `[virtual]`

Reimplemented from `gdcm::Object`.

Reimplemented in `gdcm::Image`, and `gdcm::Pixmap`.

Examples:

ExtractIconFromFile.cxx.

25.33.4.26 `void gdcm::Bitmap::SetColumns (unsigned int col)` `[inline]`

25.33.4.27 `void gdcm::Bitmap::SetDataElement (DataElement const & de)` `[inline]`

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, and iU22tomultisc.cxx.

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

Examples:

csa2img.cxx, GenFakelImage.cxx, and iU22tomultisc.cxx.

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

Examples:

CreateARGBImage.cxx, and CreateCMYKImage.cxx.

25.33.4.30 void gdcm::Bitmap::SetLossyFlag (bool *f*) [inline]

Specifically set that the image was compressed using a lossy compression mechanism.

25.33.4.31 void gdcm::Bitmap::SetLUT (LookupTable const & *lut*) [inline]

Set/Get LUT.

25.33.4.32 void gdcm::Bitmap::SetNeedByteSwap (bool *b*) [inline]

25.33.4.33 void gdcm::Bitmap::SetNumberOfDimensions (unsigned int *dim*)

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakelImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

25.33.4.34 void gdcm::Bitmap::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*)

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakelImage.cxx, and iU22tomultisc.cxx.

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

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

References gdcm::PixelFormat::Validate().

25.33.4.36 void gdcm::Bitmap::SetPlanarConfiguration (unsigned int *pc*)

Warning

you need to call SetPixelFormat first (before SetPlanarConfiguration) for consistency checking

25.33.4.37 void gdcmm::Bitmap::SetRows (unsigned int *rows*) [inline]

25.33.4.38 void gdcmm::Bitmap::SetTransferSyntax (TransferSyntax const & *ts*) [inline]

Transfer syntax.

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, and MergeTwoFiles.cxx.

25.33.4.39 bool gdcmm::Bitmap::TryJPEG2000Codec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.4.40 bool gdcmm::Bitmap::TryJPEG2000Codec2 (std::ostream & *os*) const [protected]

25.33.4.41 bool gdcmm::Bitmap::TryJPEGCodec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.4.42 bool gdcmm::Bitmap::TryJPEGCodec2 (std::ostream & *os*) const [protected]

25.33.4.43 bool gdcmm::Bitmap::TryJPEGLSCodec (char * *buffer*, bool & *lossyflag*) const [protected]

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

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

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

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

25.33.5 Friends And Related Function Documentation

25.33.5.1 friend class ImageChangeTransferSyntax [friend]

25.33.5.2 friend class PixmapReader [friend]

25.33.6 Member Data Documentation

25.33.6.1 std::vector<unsigned int> gdcmm::Bitmap::Dimensions [protected]

25.33.6.2 bool gdcmm::Bitmap::LossyFlag [protected]

25.33.6.3 LUTPtr gdcmm::Bitmap::LUT [protected]

25.33.6.4 bool gdcmm::Bitmap::NeedByteSwap [protected]

25.33.6.5 unsigned int gdcmm::Bitmap::NumberOfDimensions [protected]

25.33.6.6 PixelFormat gdcmm::Bitmap::PF [protected]

25.33.6.7 PhotometricInterpretation gdcmm::Bitmap::PI [protected]

25.33.6.8 DataElement gdcmm::Bitmap::PixelData [protected]

25.33.6.9 unsigned int gdcm::Bitmap::PlanarConfiguration [protected]

25.33.6.10 TransferSyntax gdcm::Bitmap::TS [protected]

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

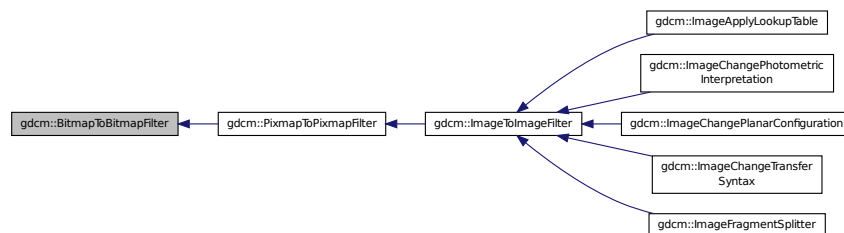
- gdcmBitmap.h

25.34 gdcm::BitmapToBitmapFilter Class Reference

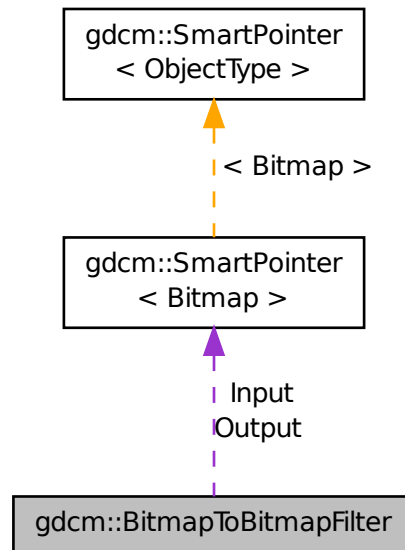
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

```
#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.
- `void SetInput (const Bitmap &image)`
Set input image.

Protected Attributes

- `SmartPointer< Bitmap > Input`
- `SmartPointer< Bitmap > Output`

25.34.1 Detailed Description

`BitmapToBitmapFilter` class Super class for all filter taking an image and producing an output image.

25.34.2 Constructor & Destructor Documentation

25.34.2.1 `gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ()`

25.34.2.2 `gdcm::BitmapToBitmapFilter::~~BitmapToBitmapFilter () [inline]`

25.34.3 Member Function Documentation

25.34.3.1 `const Bitmap& gdcm::BitmapToBitmapFilter::GetOutput () const [inline]`

Get Output image.

Reimplemented in `gdcm::ImageToImageFilter`, and `gdcm::PixmapToPixmapFilter`.

25.34.3.2 `void gdcm::BitmapToBitmapFilter::SetInput (const Bitmap & image)`

Set input image.

Examples:

`CompressImage.cxx`.

25.34.4 Member Data Documentation

25.34.4.1 `SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input [protected]`

25.34.4.2 `SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output [protected]`

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

- `gdcmBitmapToBitmapFilter.h`

25.35 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 ()`

25.35.1 Detailed Description

ByteBuffer.

Detailed description here

Note

looks like a `std::streambuf` or `std::filebuf` class with the get and peek pointer

25.35.2 Constructor & Destructor Documentation

25.35.2.1 `gdcm::ByteBuffer::ByteBuffer ()` `[inline]`

25.35.3 Member Function Documentation

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

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

25.35.3.3 `void gdcm::ByteBuffer::ShiftEnd (int len)` `[inline]`

25.35.3.4 `void gdcm::ByteBuffer::UpdatePosition ()` `[inline]`

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

- `gdcmByteBuffer.h`

25.36 `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 ()`

25.36.1 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`.

25.36.2 Member Function Documentation

25.36.2.1 `template<class T> static void gdcm::ByteSwap< T >::Swap (T & p)` `[static]`

25.36.2.2 `template<class T> static void gdcm::ByteSwap<T>::SwapFromSwapCodeIntoSystem (T & p, SwapCode const & sc) [static]`

Examples:

TestByteSwap.cxx.

25.36.2.3 `template<class T> static void gdcm::ByteSwap<T>::SwapRange (T * p, unsigned int num) [static]`

25.36.2.4 `template<class T> static void gdcm::ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem (T * p, SwapCode const & sc, std::streamoff num) [static]`

Examples:

TestByteSwap.cxx.

25.36.2.5 `template<class T> static bool gdcm::ByteSwap<T>::SystemIsBigEndian () [static]`

Query the machine Endian-ness.

25.36.2.6 `template<class T> static bool gdcm::ByteSwap<T>::SystemIsLittleEndian () [static]`

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

- gdcmByteSwap.h

25.37 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)

25.37.1 Detailed Description

ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

25.37.2 Constructor & Destructor Documentation

25.37.2.1 `gdcm::ByteSwapFilter::ByteSwapFilter (DataSet & ds)` `[inline]`

25.37.2.2 `gdcm::ByteSwapFilter::~~ByteSwapFilter ()`

25.37.3 Member Function Documentation

25.37.3.1 `bool gdcm::ByteSwapFilter::ByteSwap ()`

25.37.3.2 `void gdcm::ByteSwapFilter::SetByteSwapTag (bool b)` `[inline]`

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

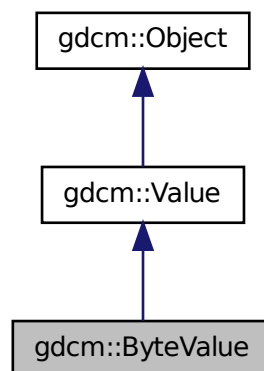
- `gdcmByteSwapFilter.h`

25.38 gdcm::ByteValue Class Reference

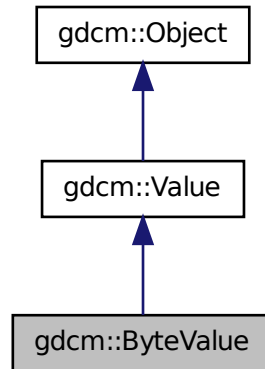
Class to represent binary value (array of bytes)

```
#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 Clear ()
- 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) / dont 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 PrintGroupLength (std::ostream &os)
- void PrintHex (std::ostream &os, VL maxlength) const
- template<typename TSwap, typename TType >
std::istream & Read (std::istream &is)
- 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

25.38.1 Detailed Description

Class to represent binary value (array of bytes)

Note

Examples:

DumpADAC.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFrom-File.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, and rle2img.cxx.

25.38.2 Constructor & Destructor Documentation

25.38.2.1 `gdcm::ByteValue::ByteValue (const char * array = 0, VL const & vl = 0)` `[inline]`

References `gdcmDebugMacro`.

25.38.2.2 `gdcm::ByteValue::ByteValue (std::vector< char > & v)` `[inline]`

Warning

casting to `uint32_t`

25.38.2.3 `gdcm::ByteValue::~~ByteValue ()` `[inline]`

25.38.3 Member Function Documentation

25.38.3.1 `void gdcm::ByteValue::Clear ()` `[inline]`, `[virtual]`

Implements `gdcm::Value`.

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

Examples:

DuplicatePCDE.cxx.

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

Examples:

FixJAIBugJPEGLS.cxx.

25.38.3.4 VL gdcm::ByteValue::GetLength () const [inline],[virtual]

Implements gdcm::Value.

Examples:

DumpADAC.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBroken-J2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), 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(), gdcm::Element< TVR, VM::VM1_n >::SetNoSwap(), and gdcm::Fragment::Write().

25.38.3.5 const char* gdcm::ByteValue::GetPointer () const [inline]

Examples:

DumpADAC.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBroken-J2K.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), 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().

25.38.3.6 bool gdcm::ByteValue::IsEmpty () const [inline]

25.38.3.7 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 dont think this function is working since it does not handle UNICODE or character set...

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

25.38.3.9 ByteValue& gdcm::ByteValue::operator= (const ByteValue & val) [inline]

25.38.3.10 bool gdcm::ByteValue::operator== (const ByteValue & val) const [inline]

25.38.3.11 bool gdcm::ByteValue::operator== (const Value & val) const [inline],[virtual]

Implements gdcm::Value.

25.38.3.12 void gdcm::ByteValue::Print (std::ostream & os) const [inline],[protected],[virtual]

Reimplemented from gdcm::Object.

25.38.3.13 void gdcM::ByteValue::PrintASCII (std::ostream & os, VL *maxlength*) const

25.38.3.14 void gdcM::ByteValue::PrintGroupLength (std::ostream & os) [inline]

25.38.3.15 void gdcM::ByteValue::PrintHex (std::ostream & os, VL *maxlength*) const

25.38.3.16 template<typename TSwap, typename TType > std::istream& gdcM::ByteValue::Read (std::istream & is)
[inline]

25.38.3.17 template<typename TSwap > std::istream& gdcM::ByteValue::Read (std::istream & is) [inline]

25.38.3.18 void gdcM::ByteValue::SetLength (VL vl) [inline],[virtual]

Implements gdcM::Value.

References gdcMDebugMacro, gdcM::VL::IsOdd(), and gdcM::VL::IsUndefined().

25.38.3.19 template<typename TSwap, typename TType > std::ostream const& gdcM::ByteValue::Write (std::ostream & os)
const [inline]

Referenced by gdcM::Fragment::Write().

25.38.3.20 template<typename TSwap > std::ostream const& gdcM::ByteValue::Write (std::ostream & os) const [inline]

25.38.3.21 bool gdcM::ByteValue::WriteBuffer (std::ostream & os) const [inline]

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

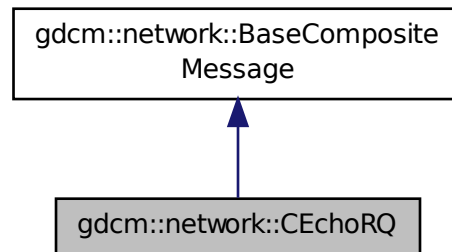
- gdcMByteValue.h

25.39 gdcM::network::CEchoRQ Class Reference

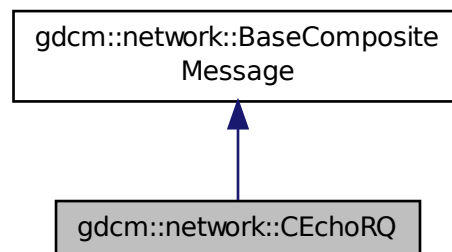
CEchoRQ this file defines the messages for the cecho action.

```
#include <gdcMCEchoMessages.h>
```

Inheritance diagram for gdcmm::network::CEchoRQ:



Collaboration diagram for gdcmm::network::CEchoRQ:



Public Member Functions

- std::vector
< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)

Public Attributes

- UIComp AffectedSOPClassUID
- uint16_t MessageID

25.39.1 Detailed Description

CEchoRQ this file defines the messages for the cecho action.

25.39.2 Member Function Documentation

25.39.2.1 `std::vector<PresentationDataValue> gdcmm::network::CEchoRQ::ConstructPDV (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [virtual]`

Implements `gdcmm::network::BaseCompositeMessage`.

25.39.3 Member Data Documentation

25.39.3.1 `UIComp gdcmm::network::CEchoRQ::AffectedSOPClassUID`

25.39.3.2 `uint16_t gdcmm::network::CEchoRQ::MessageID`

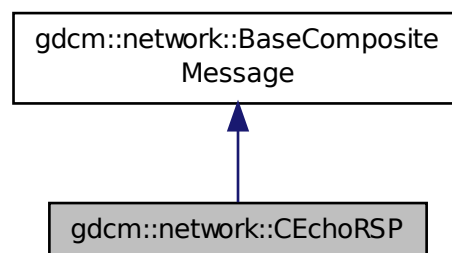
The documentation for this class was generated from the following files:

- `gdcmmCEchoMessages.h`
- `gdcmmDIMSE.h`

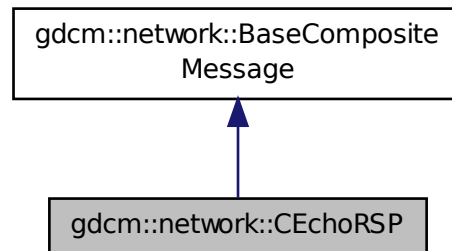
25.40 gdcmm::network::CEchoRSP Class Reference

```
#include <gdcmmCEchoMessages.h>
```

Inheritance diagram for `gdcmm::network::CEchoRSP`:



Collaboration diagram for gdcm::network::CEchoRSP:



Public Member Functions

- `std::vector`
`< PresentationDataValue > ConstructPDV (const DataSet *inDataSet)`

25.40.1 Member Function Documentation

25.40.1.1 `std::vector<PresentationDataValue> gdcm::network::CEchoRSP::ConstructPDV (const DataSet * inDataSet)`

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

- `gdcmCEchoMessages.h`

25.41 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

25.41.1 Detailed Description

PS 3.4 - 2009 Table B.2-1 C-STORE STATUS

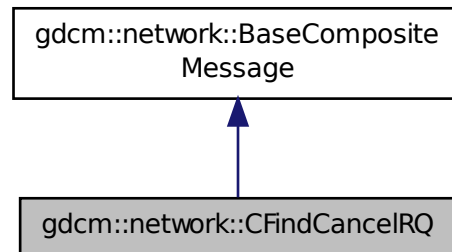
The documentation for this class was generated from the following file:

- `gdcmDIMSE.h`

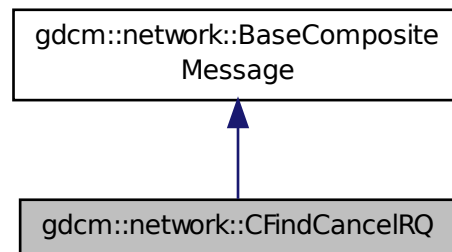
25.42 gdcm::network::CFindCancelRQ Class Reference

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for `gdcm::network::CFindCancelRQ`:



Collaboration diagram for `gdcm::network::CFindCancelRQ`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV (const DataSet *inDataSet)`

25.42.1 Member Function Documentation

25.42.1.1 `std::vector<PresentationDataValue> gdcm::network::CFindCancelRQ::ConstructPDV (const DataSet * inDataSet)`

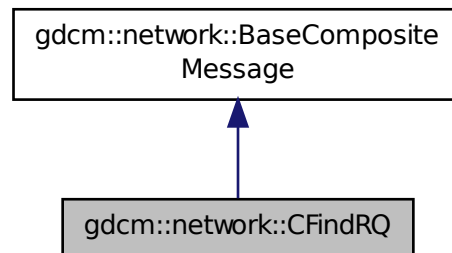
The documentation for this class was generated from the following file:

- `gdcmCFindMessages.h`

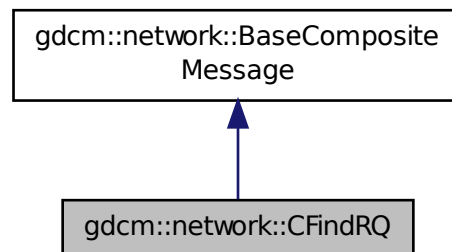
25.43 gdcmm::network::CFindRQ Class Reference

```
#include <gdcmmCFindMessages.h>
```

Inheritance diagram for gdcmm::network::CFindRQ:



Collaboration diagram for gdcmm::network::CFindRQ:



Public Member Functions

- std::vector
 < PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)

25.43.1 Member Function Documentation

25.43.1.1 `std::vector<PresentationDataValue> gdcmm::network::CFindRQ::ConstructPDV (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [virtual]`

Implements `gdcmm::network::BaseCompositeMessage`.

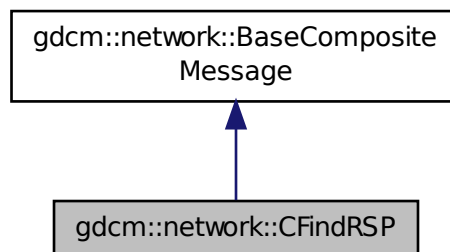
The documentation for this class was generated from the following file:

- `gdcmmCFindMessages.h`

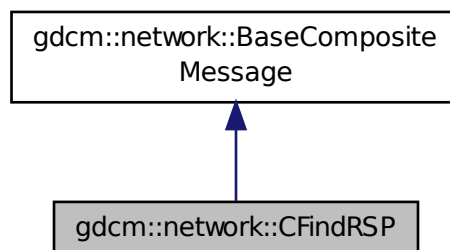
25.44 `gdcmm::network::CFindRSP` Class Reference

```
#include <gdcmmCFindMessages.h>
```

Inheritance diagram for `gdcmm::network::CFindRSP`:



Collaboration diagram for `gdcmm::network::CFindRSP`:



Public Member Functions

- `std::vector`
 `< PresentationDataValue > ConstructPDV (const DataSet *inDataSet)`

25.44.1 Member Function Documentation

25.44.1.1 `std::vector<PresentationDataValue> gdcmm::network::CFindRSP::ConstructPDV (const DataSet * inDataSet)`

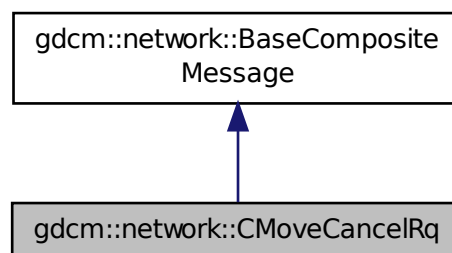
The documentation for this class was generated from the following file:

- `gdcmmCFindMessages.h`

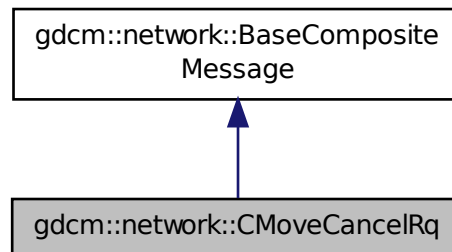
25.45 gdcmm::network::CMoveCancelRq Class Reference

```
#include <gdcmmCMoveMessages.h>
```

Inheritance diagram for `gdcmm::network::CMoveCancelRq`:



Collaboration diagram for `gdcm::network::CMoveCancelRq`:



Public Member Functions

- `std::vector`
`< PresentationDataValue > ConstructPDV (const DataSet *inDataSet)`

25.45.1 Member Function Documentation

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

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

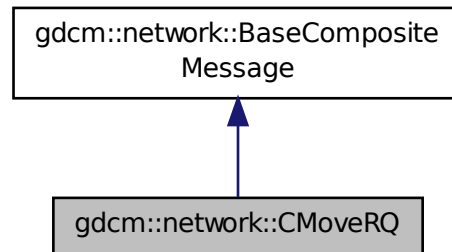
- `gdcmCMoveMessages.h`

25.46 gdcm::network::CMoveRQ Class Reference

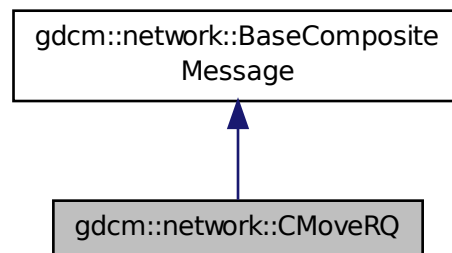
CMoveRQ this file defines the messages for the cmove action.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRQ:



Collaboration diagram for gdcm::network::CMoveRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)`

25.46.1 Detailed Description

CMoveRQ this file defines the messages for the cmove action.

25.46.2 Member Function Documentation

25.46.2.1 `std::vector<PresentationDataValue> gdcmm::network::CMoveRQ::ConstructPDV (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [virtual]`

Implements `gdcmm::network::BaseCompositeMessage`.

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

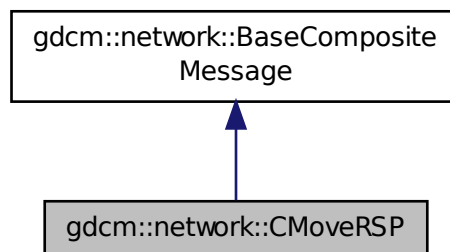
- `gdcmmCMoveMessages.h`

25.47 `gdcmm::network::CMoveRSP` Class Reference

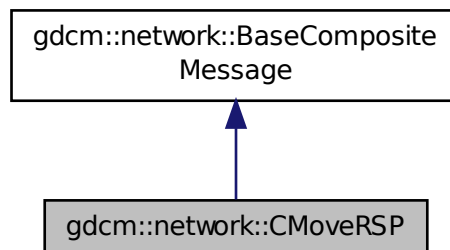
`CMoveRSP` this file defines the messages for the cmove action.

```
#include <gdcmmCMoveMessages.h>
```

Inheritance diagram for `gdcmm::network::CMoveRSP`:



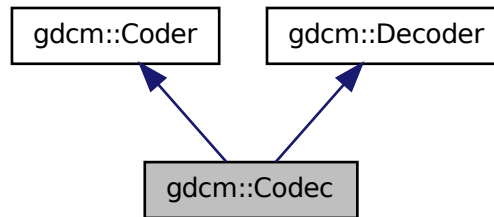
Collaboration diagram for `gdcmm::network::CMoveRSP`:



Public Member Functions

- `std::vector`
`< PresentationDataValue > ConstructPDV (const DataSet *inDataSet)`

Collaboration diagram for `gdcm::Codec`:



Additional Inherited Members

25.48.1 Detailed Description

Codec class.

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

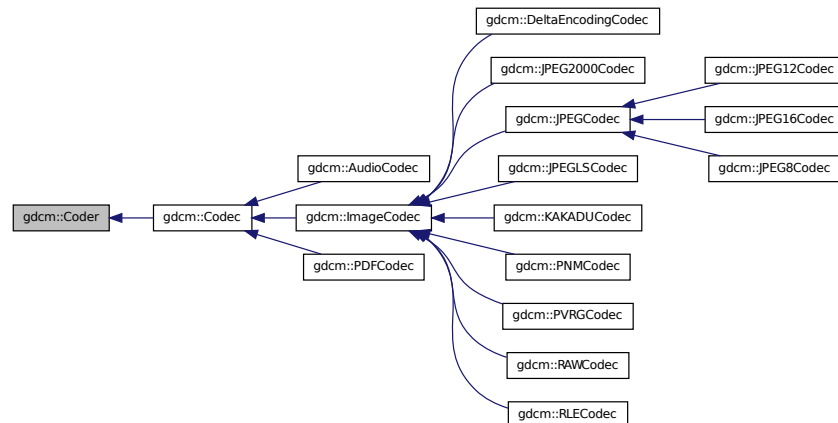
- `gdcmCodec.h`

25.49 `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)`

25.49.1 Detailed Description

Coder.

25.49.2 Constructor & Destructor Documentation

25.49.2.1 `virtual gdcm::Coder::~Coder () [inline],[virtual]`

25.49.3 Member Function Documentation

25.49.3.1 `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::PVRGCodec`, `gdcm::JPEG2000Codec`, `gdcm::RLECodec`, `gdcm::ImageCodec`, `gdcm::JPEGLSCodec`, `gdcm::PNMCodec`, `gdcm::KAKADUCodec`, `gdcm::RAWCodec`, `gdcm::AudioCodec`, and `gdcm::PDFCodec`.

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

Code.

Reimplemented in `gdcm::JPEGCodec`, `gdcm::JPEGLSCodec`, `gdcm::RLECodec`, `gdcm::PVRGCodec`, `gdcm::JPEG2000Codec`, `gdcm::KAKADUCodec`, and `gdcm::RAWCodec`.

25.49.3.3 `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`

25.50 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)

25.50.1 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.

25.50.2 Member Typedef Documentation

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

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

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

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

25.50.2.5 `typedef InternalClass::iterator gdcm::CodeString::iterator`

25.50.2.6 `typedef InternalClass::pointer gdcm::CodeString::pointer`

25.50.2.7 `typedef InternalClass::reference gdcm::CodeString::reference`

25.50.2.8 `typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator`

25.50.2.9 `typedef InternalClass::size_type gdcm::CodeString::size_type`

25.50.2.10 `typedef InternalClass::value_type gdcm::CodeString::value_type`

25.50.3 Constructor & Destructor Documentation

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

CodeString constructors.

25.50.3.2 `gdcm::CodeString::CodeString (const value_type * s) [inline]`

25.50.3.3 `gdcm::CodeString::CodeString (const value_type * s, size_type n) [inline]`

25.50.3.4 `gdcm::CodeString::CodeString (const InternalClass & s, size_type pos = 0, size_type n = InternalClass::npos) [inline]`

25.50.4 Member Function Documentation

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

Return the full code string as std::string.

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

Check if CodeString obj is correct..

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

Return the size of the string.

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

25.50.5 Friends And Related Function Documentation

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

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

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

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

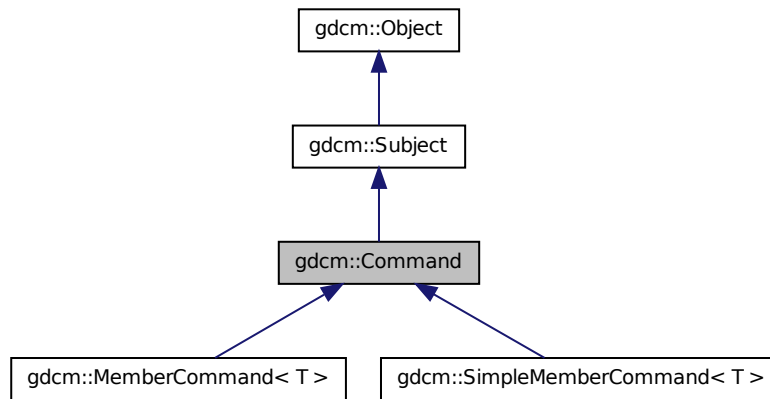
- `gdcmCodeString.h`

25.51 gdcm::Command Class Reference

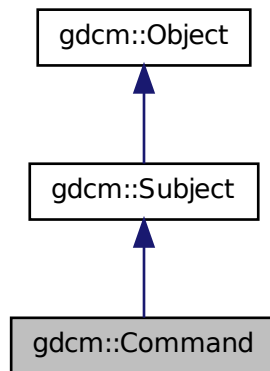
Command superclass for callback/observer methods.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::Command:



Collaboration diagram for gdcm::Command:



Public Member Functions

- 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()`

25.51.1 Detailed Description

Command superclass for callback/observer methods.

See also

Subject

25.51.2 Constructor & Destructor Documentation

25.51.2.1 `gdcM::Command::Command()` [protected]

25.51.2.2 `gdcM::Command::~~Command()` [protected]

25.51.3 Member Function Documentation

25.51.3.1 `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 >`.

25.51.3.2 `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:

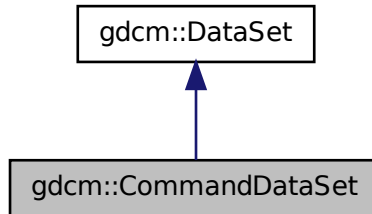
- `gdcMCommand.h`

25.52 gdcM::CommandDataSet Class Reference

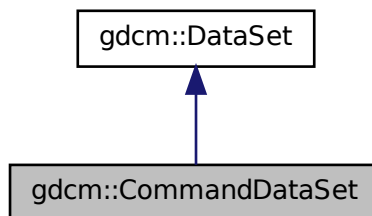
Class to represent a Command DataSet.

```
#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)
Replace a dataelement with another one.
- std::ostream & Write (std::ostream &os) const
Write.

Friends

- std::ostream & operator<< (std::ostream &_os, const CommandDataSet &_val)

Additional Inherited Members

25.52.1 Detailed Description

Class to represent a Command DataSet.

See also

DataSet

25.52.2 Constructor & Destructor Documentation

25.52.2.1 `gdcm::CommandDataSet::CommandDataSet ()` `[inline]`

25.52.2.2 `gdcm::CommandDataSet::~~CommandDataSet ()` `[inline]`

25.52.3 Member Function Documentation

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

Insert a DataElement in the DataSet.

Warning

: Tag need to be $\geq 0x8$ to be considered valid data element

Reimplemented from `gdcm::DataSet`.

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

25.52.3.2 `std::istream& gdcm::CommandDataSet::Read (std::istream & is)`

Read.

Reimplemented from `gdcm::DataSet`.

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

Replace a dataelement with another one.

Reimplemented from `gdcm::DataSet`.

References `gdcm::DataElement::GetTag()`.

25.52.3.4 `std::ostream& gdcm::CommandDataSet::Write (std::ostream & os) const`

Write.

Reimplemented from `gdcm::DataSet`.

25.52.4 Friends And Related Function Documentation

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

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

- `gdcmCommandDataSet.h`

25.53 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)`
- `static std::vector< PresentationDataValue > ConstructCStoreRSP (const DataSet *inDataSet, const BasePDU *inPC)`

25.53.1 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).

25.53.2 Member Function Documentation

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

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

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

25.53.2.4 `static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCStoreRQ (const ULConnection & inConnection, const File & file) [static]`

25.53.2.5 `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`

25.54 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, bool inMove=false)`
- `static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, bool inMove=false)`
- `static bool CStore (const char *remote, uint16_t portno, const Directory::FileNamesType &filenames, const char *aetitle=NULL, const char *call=NULL)`

25.54.1 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)

25.54.2 Member Typedef Documentation

25.54.2.1 `typedef std::vector< KeyValuePairType > gdcmm::CompositeNetworkFunctions::KeyValuePairArrayType`

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

25.54.3 Member Function Documentation

25.54.3.1 `static bool gdcmm::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' This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

25.54.3.2 `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' This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

25.54.3.3 `static bool gdcM::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.

25.54.3.4 `static BaseRootQuery* gdcM::CompositeNetworkFunctions::ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const DataSet & queryds, bool inMove = false) [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)
------------------	--

25.54.3.5 `static BaseRootQuery* gdcM::CompositeNetworkFunctions::ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType & keys, bool inMove = false) [static]`

Deprecated

25.54.3.6 `static bool gdcM::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' This is an error to set remote to NULL or portno to 0

Returns

true if it worked for all files

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

- gdcmCompositeNetworkFunctions.h

25.55 gdcm::ConstCharWrapper Class Reference

Do not use me.

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

Public Member Functions

- ConstCharWrapper (const char *i=0)
- operator const char * () const

25.55.1 Detailed Description

Do not use me.

25.55.2 Constructor & Destructor Documentation

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

25.55.3 Member Function Documentation

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

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

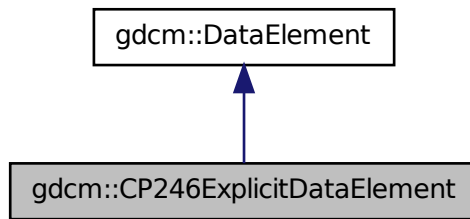
- gdcmConstCharWrapper.h

25.56 gdcm::CP246ExplicitDataElement Class Reference

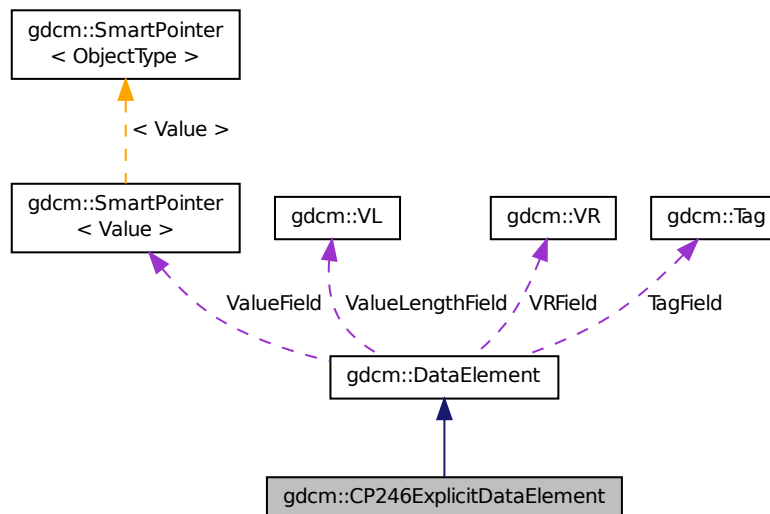
Class to read/write a DataElement as CP246Explicit Data Element.

```
#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)`
- `template<typename TSwap >`
`std::istream & ReadWithLength (std::istream &is, VL &length)`

Additional Inherited Members

25.56.1 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

25.56.2 Member Function Documentation

25.56.2.1 VL gdcM::CP246ExplicitDataElement::GetLength () const

Reimplemented from gdcM::DataElement.

25.56.2.2 template<typename TSwap > std::istream& gdcM::CP246ExplicitDataElement::Read (std::istream & is)

Reimplemented from gdcM::DataElement.

25.56.2.3 template<typename TSwap > std::istream& gdcM::CP246ExplicitDataElement::ReadPreValue (std::istream & is)

25.56.2.4 template<typename TSwap > std::istream& gdcM::CP246ExplicitDataElement::ReadValue (std::istream & is)

25.56.2.5 template<typename TSwap > std::istream& gdcM::CP246ExplicitDataElement::ReadWithLength (std::istream & is, VL & length)

Reimplemented from gdcM::DataElement.

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

- gdcMCP246ExplicitDataElement.h

25.57 gdcM::CryptographicMessageSyntax Class Reference

Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

```
#include <gdcMCryptographicMessageSyntax.h>
```

Public Types

- enum CipherTypes {
 DES_CIPHER,
 DES3_CIPHER,
 AES128_CIPHER,
 AES192_CIPHER,
 AES256_CIPHER }

Public Member Functions

- `CryptographicMessageSyntax ()`
- `~CryptographicMessageSyntax ()`
- `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)`

25.57.1 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

25.57.2 Member Enumeration Documentation

25.57.2.1 enum `gdcmm::CryptographicMessageSyntax::CipherTypes`

Enumerator:

DES_CIPHER
DES3_CIPHER
AES128_CIPHER
AES192_CIPHER
AES256_CIPHER

25.57.3 Constructor & Destructor Documentation

25.57.3.1 `gdcmm::CryptographicMessageSyntax::CryptographicMessageSyntax ()`

25.57.3.2 `gdcmm::CryptographicMessageSyntax::~~CryptographicMessageSyntax ()`

25.57.4 Member Function Documentation

25.57.4.1 `bool gdcmm::CryptographicMessageSyntax::Decrypt (char * output, size_t & outlen, const char * array, size_t len) const`

decrypt content from a PKCS#7 envelopedData structure

25.57.4.2 `bool gdcmm::CryptographicMessageSyntax::Encrypt (char * output, size_t & outlen, const char * array, size_t len) const`

create a PKCS#7 envelopedData structure

25.57.4.3 **CipherTypes** gdcM::CryptographicMessageSyntax::GetCipherType () const

25.57.4.4 bool gdcM::CryptographicMessageSyntax::ParseCertificateFile (const char * *filename*)

25.57.4.5 bool gdcM::CryptographicMessageSyntax::ParseKeyFile (const char * *filename*)

25.57.4.6 void gdcM::CryptographicMessageSyntax::SetCipherType (CipherTypes *type*)

Set Cipher Type. Default is: AES256_CIPHER

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

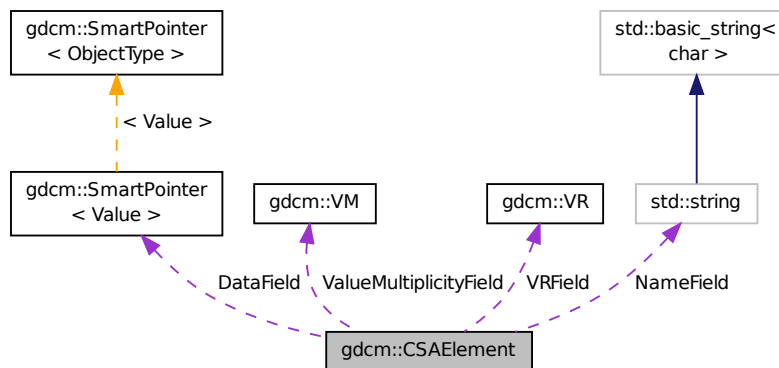
- gdcMCryptographicMessageSyntax.h

25.58 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)

25.58.1 Detailed Description

Class to represent a CSA Element.

See also

CSAHeader

Examples:

csa2img.cxx, and MrProtocol.cxx.

25.58.2 Member Typedef Documentation

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

25.58.3 Constructor & Destructor Documentation

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

25.58.3.2 `gdcmm::CSAElement::CSAElement (const CSAElement & _val)` [inline]

25.58.4 Member Function Documentation

25.58.4.1 `const ByteValue* gdcmm::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.

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

Set/Get Key.

Referenced by operator<().

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

Set/Get Name.

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

Set/Get NoOfItems.

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

Set/Get SyngoDT.

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

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

Examples:

csa2img.cxx.

25.58.4.7 **Value& gdcM::CSAElement::GetValue ()** [inline]

25.58.4.8 **const VM& gdcM::CSAElement::GetVM () const** [inline]

Set/Get VM.

25.58.4.9 **VR const& gdcM::CSAElement::GetVR () const** [inline]

Set/Get VR.

25.58.4.10 **bool gdcM::CSAElement::IsEmpty () const** [inline]

Check if CSA Element is empty.

Examples:

csa2img.cxx.

25.58.4.11 **bool gdcM::CSAElement::operator< (const CSAElement & de) const** [inline]

References GetKey().

25.58.4.12 **CSAElement& gdcM::CSAElement::operator= (const CSAElement & de)** [inline]

References DataField, KeyField, NameField, NoOfItemsField, SyngoDTField, ValueMultiplicityField, and VRField.

25.58.4.13 **bool gdcM::CSAElement::operator== (const CSAElement & de) const** [inline]

References KeyField, NameField, SyngoDTField, ValueMultiplicityField, and VRField.

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

Set.

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

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

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

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

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

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

25.58.4.21 **void gdcM::CSAElement::SetVR (VR const & vr)** [inline]

25.58.5 Friends And Related Function Documentation

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

25.58.6 Member Data Documentation

25.58.6.1 `DataPtr gdcm::CSAElement::DataField` [protected]

Referenced by `gdcm::operator<<()`, and `operator=()`.

25.58.6.2 `unsigned int gdcm::CSAElement::KeyField` [protected]

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

25.58.6.3 `std::string gdcm::CSAElement::NameField` [protected]

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

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

Referenced by `gdcm::operator<<()`, and `operator=()`.

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

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

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

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

25.58.6.7 `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`

25.59 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 & GetCSAEnd () const

Friends

- std::ostream & operator<< (std::ostream &_os, const CSAHeader &d)

25.59.1 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.

25.59.2 Member Enumeration Documentation

25.59.2.1 enum gdcm::CSAHeader::CSAHeaderType

Divers format of CSAHeader as found 'in the wild'.

Enumerator:

UNKNOWN

SV10

NOMAGIC

DATASET_FORMAT

INTERFILE

ZEROED_OUT

25.59.3 Constructor & Destructor Documentation

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

25.59.3.2 gdcm::CSAHeader::~~CSAHeader () [inline]

25.59.4 Member Function Documentation

25.59.4.1 `bool gdcM::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.

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

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

25.59.4.3 `const CSAElement& gdcM::CSAHeader::GetCSAEnd () const [protected]`

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

Return the CSAElement corresponding to name 'name'

Warning

Case Sensitive

Examples:

csa2img.cxx, and MrProtocol.cxx.

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

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

Examples:

csa2img.cxx, and PublicDict.cxx.

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

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

Examples:

MrProtocol.cxx.

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

Return the DataSet output (use only if Format == DATASET_FORMAT)

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

return the format of the CSAHeader SV10 and NOMAGIC are equivalent.

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

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

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

Decode the CSAHeader from element 'de'.

Examples:

csa2img.cxx, and MrProtocol.cxx.

25.59.4.11 `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<<()`.

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

25.59.4.13 `template<typename TSwap > const std::ostream& gdcm::CSAHeader::Write (std::ostream & os) const`

25.59.5 Friends And Related Function Documentation

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

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

- `gdcmCSAHeader.h`

25.60 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)

25.60.1 Detailed Description

Class to represent a map of CSAHeaderDictEntry.

Examples:

MrProtocol.cxx.

25.60.2 Member Typedef Documentation

25.60.2.1 typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator

25.60.2.2 typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator

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

25.60.3 Constructor & Destructor Documentation

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

25.60.4 Member Function Documentation

25.60.4.1 void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (const CSAHeaderDictEntry & de) [inline]

25.60.4.2 ConstIterator gdcm::CSAHeaderDict::Begin () const [inline]

25.60.4.3 ConstIterator gdcm::CSAHeaderDict::End () const [inline]

25.60.4.4 const CSAHeaderDictEntry& gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (const char * name) const [inline]

Examples:

MrProtocol.cxx.

25.60.4.5 bool gdcm::CSAHeaderDict::IsEmpty () const [inline]

25.60.4.6 void gdcm::CSAHeaderDict::LoadDefault () [protected]

25.60.5 Friends And Related Function Documentation

25.60.5.1 friend class Dicts [friend]

25.60.5.2 std::ostream& operator<< (std::ostream & _os, const CSAHeaderDict & _val) [friend]

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

- gdcmCSAHeaderDict.h

25.61 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)

25.61.1 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.

25.61.2 Constructor & Destructor Documentation

25.61.2.1 gdcM::CSAHeaderDictEntry::CSAHeaderDictEntry (const char * *name* = " ", VR const & *vr* = VR::INVALID, VM const & *vm* = VM::VMO, const char * *desc* = " ") [inline]

25.61.3 Member Function Documentation

25.61.3.1 const char* gdcM::CSAHeaderDictEntry::GetDescription () const [inline]

Set/Get Description.

25.61.3.2 const char* gdcM::CSAHeaderDictEntry::GetName () const [inline]

Set/Get Name.

Referenced by operator<().

25.61.3.3 const VM& gdcM::CSAHeaderDictEntry::GetVM () const [inline]

Set/Get VM.

25.61.3.4 const VR& gdcM::CSAHeaderDictEntry::GetVR () const [inline]

Set/Get VR.

25.61.3.5 `bool gdcm::CSAHeaderDictEntry::operator< (const CSAHeaderDictEntry & entry) const` `[inline]`

References `GetName()`.

25.61.3.6 `void gdcm::CSAHeaderDictEntry::SetDescription (const char * desc)` `[inline]`

25.61.3.7 `void gdcm::CSAHeaderDictEntry::SetName (const char * name)` `[inline]`

25.61.3.8 `void gdcm::CSAHeaderDictEntry::SetVM (VM const & vm)` `[inline]`

25.61.3.9 `void gdcm::CSAHeaderDictEntry::SetVR (const VR & vr)` `[inline]`

25.61.4 Friends And Related Function Documentation

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

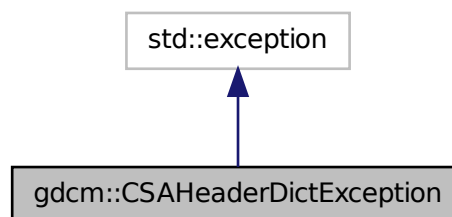
The documentation for this class was generated from the following file:

- `gdcmCSAHeaderDictEntry.h`

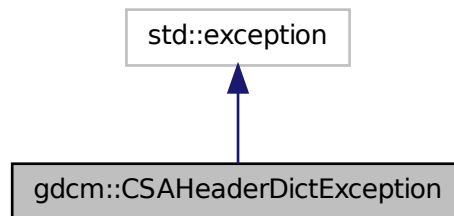
25.62 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for `gdcm::CSAHeaderDictException`:



Collaboration diagram for `gdcm::CSAHeaderDictException`:



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

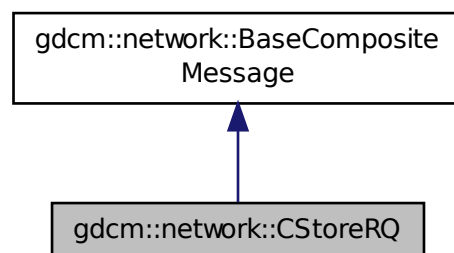
- `gdcmCSAHeaderDict.h`

25.63 `gdcm::network::CStoreRQ` Class Reference

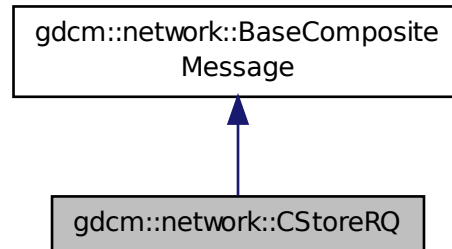
`CStoreRQ` this file defines the messages for the cecho action.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for `gdcm::network::CStoreRQ`:



Collaboration diagram for gdcm::network::CStoreRQ:



Public Member Functions

- `std::vector`
`< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const File &file)`

25.63.1 Detailed Description

CStoreRQ this file defines the messages for the cecho action.

25.63.2 Member Function Documentation

25.63.2.1 `std::vector<PresentationDataValue> gdcm::network::CStoreRQ::ConstructPDV (const ULConnection & inConnection, const File & file)`

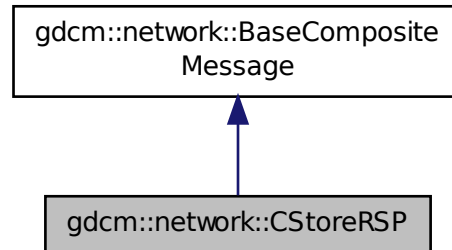
The documentation for this class was generated from the following file:

- `gdcmCStoreMessages.h`

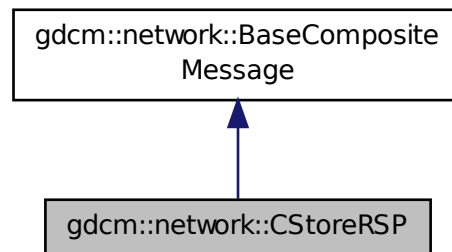
25.64 gdcm::network::CStoreRSP Class Reference

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for `gdcm::network::CStoreRSP`:



Collaboration diagram for `gdcm::network::CStoreRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV (const DataSet *inDataSet, const BasePDU *inPC)`

25.64.1 Member Function Documentation

25.64.1.1 `std::vector<PresentationDataValue> gdcm::network::CStoreRSP::ConstructPDV (const DataSet * inDataSet, const BasePDU * inPC)`

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

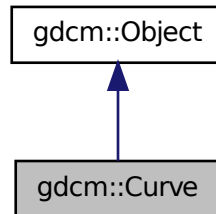
- `gdcmCStoreMessages.h`

25.65 gdcm::Curve Class Reference

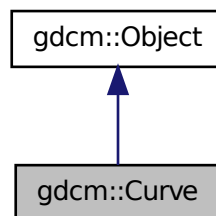
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

```
#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
- 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

25.65.1 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`

25.65.2 Constructor & Destructor Documentation

25.65.2.1 `gdcm::Curve::Curve ()`

25.65.2.2 `gdcm::Curve::~~Curve ()`

25.65.2.3 `gdcm::Curve::Curve (Curve const & ov)`

25.65.3 Member Function Documentation

25.65.3.1 `void gdcm::Curve::Decode (std::istream & is, std::ostream & os)`

25.65.3.2 `void gdcm::Curve::GetAsPoints (float * array) const`

- 25.65.3.3 unsigned short gdcm::Curve::GetDataValueRepresentation () const
- 25.65.3.4 unsigned short gdcm::Curve::GetDimensions () const
- 25.65.3.5 unsigned short gdcm::Curve::GetGroup () const
- 25.65.3.6 static unsigned int gdcm::Curve::GetNumberOfCurves (DataSet const & ds) [static]
- 25.65.3.7 unsigned short gdcm::Curve::GetNumberOfPoints () const
- 25.65.3.8 const char* gdcm::Curve::GetTypeOfData () const
- 25.65.3.9 const char* gdcm::Curve::GetTypeOfDataDescription () const
- 25.65.3.10 bool gdcm::Curve::IsEmpty () const
- 25.65.3.11 void gdcm::Curve::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Object.

- 25.65.3.12 void gdcm::Curve::SetCoordinateStartValue (unsigned short v)
- 25.65.3.13 void gdcm::Curve::SetCoordinateStepValue (unsigned short v)
- 25.65.3.14 void gdcm::Curve::SetCurve (const char * array, unsigned int length)
- 25.65.3.15 void gdcm::Curve::SetCurveDataDescriptor (const uint16_t * values, size_t num)
- 25.65.3.16 void gdcm::Curve::SetCurveDescription (const char * curvedescription)
- 25.65.3.17 void gdcm::Curve::SetDataValueRepresentation (unsigned short datavaluerepresentation)
- 25.65.3.18 void gdcm::Curve::SetDimensions (unsigned short dimensions)
- 25.65.3.19 void gdcm::Curve::SetGroup (unsigned short group)
- 25.65.3.20 void gdcm::Curve::SetNumberOfPoints (unsigned short numberofpoints)
- 25.65.3.21 void gdcm::Curve::SetTypeOfData (const char * typeofdata)
- 25.65.3.22 void gdcm::Curve::Update (const DataElement & de)

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

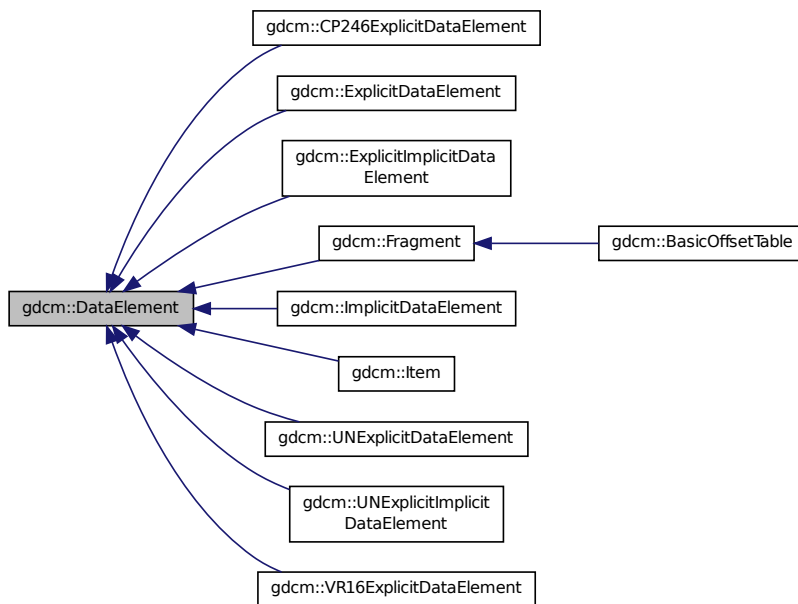
- gdcmCurve.h

25.66 gdcm::DataElement Class Reference

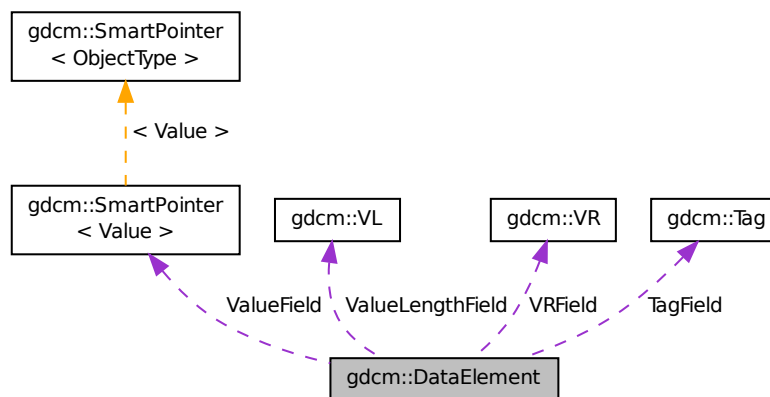
Class to represent a Data Element either Implicit or Explicit.

```
#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
- const SequenceOfItems * GetSequenceOfItems () const
- SequenceOfItems * GetSequenceOfItems ()
- 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 & 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 Attributes

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

Friends

- `std::ostream & operator<< (std::ostream &_os, const DataElement &_val)`

25.66.1 Detailed Description

Class to represent a Data Element either Implicit or Explicit.

DATA ELEMENT: A unit of information as defined by a single entry in the data dictionary. An encoded Information Object Definition (IOD) Attribute that is composed of, at a minimum, three fields: a Data Element Tag, a Value Length, and a Value Field. For some specific Transfer Syntaxes, a Data Element also contains a VR Field where the Value Representation of that Data Element is specified explicitly.

Design:

- A DataElement in GDCM always store VL (Value Length) on a 32 bits integer even when VL is 16 bits
- A DataElement always store the VR even for Implicit TS, in which case VR is defaulted to VR::INVALID
- For Item start/end (See 0xfffe tags), Value is NULL

See also

ExplicitDataElement ImplicitDataElement

Examples:

ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, Extract-EncryptedContent.cxx, ExtractIconFromFile.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtonplan.cxx, gdcmrtpian.cxx, GenAllVR.cxx, GenFakelIdentifyFile.cxx, GenFakelImage.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.

25.66.2 Member Typedef Documentation

25.66.2.1 `typedef SmartPointer<Value> gdcm::DataElement::ValuePtr` [protected]

25.66.3 Constructor & Destructor Documentation

25.66.3.1 `gdcm::DataElement::DataElement (const Tag & t = Tag (0), const VL & vl = 0, const VR & vr = VR::INVALID)`
[inline]

25.66.3.2 `gdcm::DataElement::DataElement (const DataElement & _val) [inline]`

25.66.4 Member Function Documentation

25.66.4.1 `void gdcm::DataElement::Clear () [inline]`

Clear Data Element (make Value empty and invalidate Tag & VR)

Reimplemented in `gdcm::Item`.

References `gdcm::VR::INVALID`.

Referenced by `gdcm::Item::Clear()`.

25.66.4.2 `void gdcm::DataElement::Empty () [inline]`

Make Data Element empty (no Value)

25.66.4.3 `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, 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()`.

25.66.4.4 `template<typename TDE> VL gdcm::DataElement::GetLength () const [inline]`

Reimplemented in `gdcm::Item`, `gdcm::Fragment`, `gdcm::UNExplicitImplicitDataElement`, `gdcm::ExplicitImplicitDataElement`, `gdcm::VR16ExplicitDataElement`, `gdcm::CP246ExplicitDataElement`, `gdcm::ImplicitDataElement`, `gdcm::UNExplicitDataElement`, and `gdcm::ExplicitDataElement`.

25.66.4.5 `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.

25.66.4.6 **const SequenceOfItems*** gdcm::DataElement::GetSequenceOfItems () const

Return the Value of DataElement as a Sequence Of Items (if possible)

Warning

: You need to check for NULL return value
 : In some case a Value could not have been recognized as a SequenceOfItems in those case the return of the function will be NULL, while the Value would be a valid SequenceOfItems, in those case prefer GetValueAsSQ. In which case the code internally trigger an assert to warn developer. When in doubt do not use this function and prefer GetValueAsSQ()

Deprecated Replaced by DataElement::GetValueAsSQ() as of GDCM 2.2.

25.66.4.7 **SequenceOfItems*** gdcm::DataElement::GetSequenceOfItems ()

25.66.4.8 **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::SequenceOfFragments::Read(), gdcm::SequenceOfItems::Read(), 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().

25.66.4.9 **Tag&** gdcm::DataElement::GetTag () [inline]

25.66.4.10 **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().

25.66.4.11 **Value&** gdcm::DataElement::GetValue () [inline]

25.66.4.12 **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, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

25.66.4.13 `const VL& gdcm::DataElement::GetVL () const` `[inline]`

Get VL.

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::SequenceOfFragments::Read()`, and `gdcm::SequenceOfItems::Read()`.

25.66.4.14 `VL& gdcm::DataElement::GetVL ()` `[inline]`

25.66.4.15 `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::Element< TVR, VM::VM1_n >::GetAsDataElement()`, `gdcm::Attribute< Group, 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()`.

25.66.4.16 `bool gdcm::DataElement::IsEmpty () const` `[inline]`

Check if Data Element is empty.

Examples:

DumpADAC.cxx, DumpGEMSMovieGroup.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()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet()`.

25.66.4.17 `bool gdcm::DataElement::IsUndefinedLength () const` `[inline]`

return if Value Length if of undefined length

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

References `GetTag()`.

25.66.4.19 `DataElement& gdcm::DataElement::operator= (const DataElement & de)` `[inline]`

References `TagField`, `ValueField`, `ValueLengthField`, and `VRField`.

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

References `TagField`, `ValueField`, `ValueLengthField`, and `VRField`.

25.66.4.21 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::Read (std::istream & is)`
`[inline]`

Reimplemented in `gdcm::Item`, `gdcm::Fragment`, `gdcm::BasicOffsetTable`, `gdcm::UNExplicitImplicitDataElement`, `gdcm::ExplicitImplicitDataElement`, `gdcm::VR16ExplicitDataElement`, `gdcm::CP246ExplicitDataElement`, `gdcm::ImplicitDataElement`, `gdcm::UNExplicitDataElement`, and `gdcm::ExplicitDataElement`.

25.66.4.22 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadOrSkip (std::istream & is, std::set< Tag > const & skiptags)` `[inline]`

25.66.4.23 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadPreValue (std::istream & is, std::set< Tag > const & skiptags)` `[inline]`

25.66.4.24 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadValue (std::istream & is, std::set< Tag > const & skiptags)` `[inline]`

25.66.4.25 `template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadWithLength (std::istream & is, VL & length)` `[inline]`

Reimplemented in `gdcm::ExplicitImplicitDataElement`, `gdcm::VR16ExplicitDataElement`, `gdcm::CP246ExplicitDataElement`, `gdcm::ImplicitDataElement`, `gdcm::UNExplicitDataElement`, and `gdcm::ExplicitDataElement`.

25.66.4.26 `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:

`ChangeSequenceUltrasound.cxx`, `CreateARGBImage.cxx`, `CreateCMYKImage.cxx`, `Fake_Image_Using_Stream_Image_Writer.cxx`, `FixBrokenJ2K.cxx`, `FixJAIBugJPEGLS.cxx`, `GenFakeIdentifyFile.cxx`, `GenFakeImage.cxx`, `GenLongSeqs.cxx`, `GenSeqs.cxx`, `GetSubSequenceData.cxx`, `iU22tomultisc.cxx`, and `StreamImageReaderTest.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::Element< TVR, VM::VM1_n >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, and `gdcm::SequenceOfFragments::Read()`.

25.66.4.27 `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`.

25.66.4.28 `void gdcm::DataElement::SetValue (Value const & v) [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()`.

25.66.4.29 `void gdcm::DataElement::SetVL (const VL & vl) [inline]`

Set VL Use with cautious (need to match Part 6), advanced user only

See also

`SetByteValue`

25.66.4.30 `void gdcm::DataElement::SetVLToUndefined ()`

Examples:

`Fake_Image_Using_Stream_Image_Writer.cxx`, `GenAllIVR.cxx`, `GenFakeIdentifyFile.cxx`, `GenLongSeqs.cxx`, and `GenSeqs.cxx`.

25.66.4.31 `void gdcm::DataElement::SetVR (VR const & vr) [inline]`

Set VR Use with cautious (need to match Part 6), advanced user only

Precondition

`vr` is a `VR::VRALL` (not a dual one such as `OB_OW`)

Examples:

`Fake_Image_Using_Stream_Image_Writer.cxx`, `FixBrokenJ2K.cxx`, `FixJAIBugJPEGLS.cxx`, `GenFakeIdentifyFile.cxx`, `GenLongSeqs.cxx`, `GenSeqs.cxx`, `GetSubSequenceData.cxx`, `iU22tomultisc.cxx`, and `StreamImageReader-Test.cxx`.

References `gdcm::VR::IsVRFile()`.

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::Element< TVR, VM::VM1_n >::GetAsDataElement()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`.

25.66.4.32 `template<typename TDE , typename TSwap > const std::ostream& gdcm::DataElement::Write (std::ostream & os)`
`const [inline]`

Reimplemented in `gdcm::Item`, `gdcm::Fragment`, `gdcm::ImplicitDataElement`, and `gdcm::ExplicitDataElement`.

25.66.5 Friends And Related Function Documentation

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

25.66.6 Member Data Documentation

25.66.6.1 `Tag gdcm::DataElement::TagField` `[protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

25.66.6.2 `ValuePtr gdcm::DataElement::ValueField` `[protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

25.66.6.3 `VL gdcm::DataElement::ValueLengthField` `[protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

25.66.6.4 `VR gdcm::DataElement::VRField` `[protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

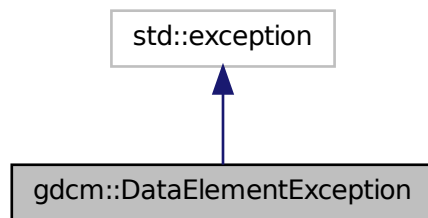
The documentation for this class was generated from the following file:

- `gdcmDataElement.h`

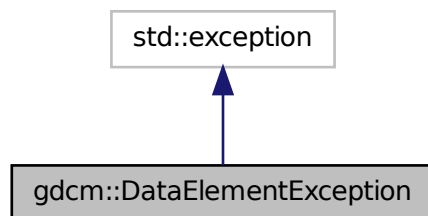
25.67 gdcm::DataElementException Class Reference

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



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

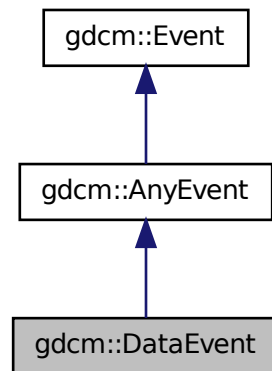
- gdcmDataSet.h

25.68 gdcm::DataEvent Class Reference

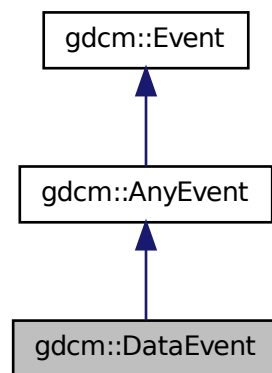
DataEvent.

```
#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)

25.68.1 Detailed Description

DataEvent.

25.68.2 Member Typedef Documentation

25.68.2.1 typedef DataEvent gdcm::DataEvent::Self

25.68.2.2 typedef AnyEvent gdcm::DataEvent::Superclass

25.68.3 Constructor & Destructor Documentation

25.68.3.1 gdcm::DataEvent::DataEvent (const char * bytes = 0, size_t len = 0) [inline]

25.68.3.2 virtual gdcm::DataEvent::~~DataEvent () [inline],[virtual]

25.68.3.3 gdcm::DataEvent::DataEvent (const Self & s) [inline]

25.68.4 Member Function Documentation

25.68.4.1 virtual bool gdcm::DataEvent::CheckEvent (const ::gdcm::Event * e) const [inline],[virtual]

25.68.4.2 const char* gdcm::DataEvent::GetData () const [inline]

25.68.4.3 size_t gdcm::DataEvent::GetDataLength () const [inline]

25.68.4.4 virtual const char* gdcm::DataEvent::GetEventName () const [inline],[virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

25.68.4.5 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.

25.68.4.6 void gdcm::DataEvent::SetData (const char * bytes, size_t len) [inline]

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

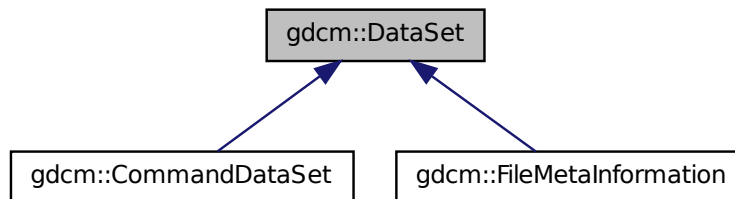
- gdcmDataEvent.h

25.69 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
DataElementSet::const_iterator ConstIterator
- typedef std::set< DataElement > DataElementSet
- typedef DataElementSet::iterator Iterator
- typedef DataElementSet::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 DataElementSet & GetDES () const

- DataSet & GetDES ()
- template<typename TDE >
VL GetLength () 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 & ReadSelectedTags (std::istream &is, const std::set< Tag > &tags)
- template<typename TDE , typename TSwap >
std::istream & ReadSelectedTagsWithLength (std::istream &is, const std::set< Tag > &tags, VL &length)
- 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, 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)

25.69.1 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:

ChangeSequenceUltrasound.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, Dump-GEMSMovieGroup.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, EncapsulateFileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtonplan.cxx, gdcmrtpplan.cxx, GenAllIVR.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.

25.69.2 Member Typedef Documentation

25.69.2.1 `typedef DataElementSet::const_iterator gdcmm::DataSet::ConstIterator`

25.69.2.2 `typedef std::set<DataElement> gdcmm::DataSet::DataElementSet`

25.69.2.3 `typedef DataElementSet::iterator gdcmm::DataSet::Iterator`

25.69.2.4 `typedef DataElementSet::size_type gdcmm::DataSet::SizeType`

25.69.3 Member Function Documentation

25.69.3.1 `ConstIterator gdcmm::DataSet::Begin () const` `[inline]`

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

25.69.3.2 `Iterator gdcmm::DataSet::Begin ()` `[inline]`

25.69.3.3 `void gdcmm::DataSet::Clear ()` `[inline]`

Referenced by `gdcmm::Item::Read()`.

25.69.3.4 **Tag** gdcm::DataSet::ComputeDataElement (const PrivateTag & t) const [protected]

25.69.3.5 **template**<typename TDE > unsigned int gdcm::DataSet::ComputeGroupLength (Tag const & tag) const [inline]

References gdcm::Tag::GetElement(), and gdcm::Tag::GetGroup().

25.69.3.6 **ConstIterator** gdcm::DataSet::End () const [inline]

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

25.69.3.7 **Iterator** gdcm::DataSet::End () [inline]

25.69.3.8 **bool** gdcm::DataSet::FindDataElement (const PrivateTag & t) const

Look up if private tag 't' is present in the dataset:

Examples:

ChangeSequenceUltrasound.cxx, csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.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(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet().

25.69.3.9 **bool** gdcm::DataSet::FindDataElement (const Tag & t) const [inline]

25.69.3.10 **const DataElement&** gdcm::DataSet::FindNextDataElement (const Tag & t) const [inline]

Examples:

DuplicatePCDE.cxx.

25.69.3.11 **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:

ChangeSequenceUltrasound.cxx, csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, ELSCINT1-WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, i-U22tomultisc.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, TVM >::SetFromDataSet()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet()`.

25.69.3.12 `const DataElement& gdcmm::DataSet::GetDataElement (const PrivateTag & t) const`

Return the dataelement.

25.69.3.13 `const DataElement& gdcmm::DataSet::GetDEEnd () const` `[protected]`

25.69.3.14 `const DataElementSet& gdcmm::DataSet::GetDES () const` `[inline]`

Examples:

`ReadAndDumpDICOMDIR.cxx`.

25.69.3.15 `DataElementSet& gdcmm::DataSet::GetDES ()` `[inline]`

25.69.3.16 `template<typename TDE> VL gdcmm::DataSet::GetLength () const` `[inline]`

25.69.3.17 `std::string gdcmm::DataSet::GetPrivateCreator (const Tag & t) const`

Return the private creator of the private tag 't':

Examples:

`DuplicatePCDE.cxx`.

25.69.3.18 `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

Reimplemented in `gdcmm::FileMetaInformation`, and `gdcmm::CommandDataSet`.

Examples:

`CreateJIPIDDataSet.cxx`, `DuplicatePCDE.cxx`, `Extracting_All_Resolution.cxx`, `Fake_Image_Using_Stream_Image_Writer.cxx`, `GenAllVR.cxx`, `GenFakeIdentifyFile.cxx`, `GenLongSeqs.cxx`, `GenSeqs.cxx`, and `StreamImageReader-Test.cxx`.

References `gdcmmErrorMacro`, `gdcmm::Tag::GetGroup()`, and `gdcmm::DataElement::GetTag()`.

25.69.3.19 `void gdcmm::DataSet::InsertDataElement (const DataElement & de)` `[inline]`, `[protected]`

References `gdcmmWarningMacro`, `gdcmm::Value::GetLength()`, `gdcmm::DataElement::GetValue()`, `gdcmm::DataElement::GetVL()`, and `gdcmm::DataElement::IsEmpty()`.

25.69.3.20 `bool gdcm::DataSet::IsEmpty () const [inline]`

Returns if the dataset is empty.

Referenced by `gdcm::Item::Read()`.

25.69.3.21 `const DataElement& gdcm::DataSet::operator() (uint16_t group, uint16_t element) const [inline]`

25.69.3.22 `DataSet& gdcm::DataSet::operator= (DataSet const & val) [inline]`

25.69.3.23 `const DataElement& gdcm::DataSet::operator[] (const Tag & t) const [inline]`

25.69.3.24 `void gdcm::DataSet::Print (std::ostream & os, std::string const & indent = " ") const [inline]`

Referenced by `gdcm::operator<<()`.

25.69.3.25 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::Read (std::istream & is)`

Reimplemented in `gdcm::FileMetaInformation`, and `gdcm::CommandDataSet`.

25.69.3.26 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadNested (std::istream & is)`

25.69.3.27 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTags (std::istream & is,
const std::set< Tag > & tags)`

25.69.3.28 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTagsWithLength (
std::istream & is, const std::set< Tag > & tags, VL & length)`

25.69.3.29 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTag (std::istream & is, const
Tag & t, std::set< Tag > const & skiptags)`

25.69.3.30 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTagWithLength (std::istream &
is, const Tag & t, VL & length)`

25.69.3.31 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadWithLength (std::istream & is, VL &
length)`

25.69.3.32 `SizeType gdcm::DataSet::Remove (const Tag & tag) [inline]`

Completely remove a dataelement from the dataset.

Examples:

`GenFakeldentifyFile.cxx`, `LargeVRDSExplicit.cxx`, `MergeTwoFiles.cxx`, `pmsct_rgb1.cxx`, and `rle2img.cxx`.

25.69.3.33 `void gdcm::DataSet::Replace (const DataElement & de) [inline]`

Replace a dataelement with another one.

Reimplemented in `gdcm::FileMetaInformation`, and `gdcm::CommandDataSet`.

Examples:

ChangeSequenceUltrasound.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeIdentifyFile.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.69.3.34 `void gdcm::DataSet::ReplaceEmpty (const DataElement & de) [inline]`

Only replace a DICOM attribute when it is missing or empty.

25.69.3.35 `SizeType gdcm::DataSet::Size () const [inline]`

Examples:

DumpGEMSMovieGroup.cxx.

Referenced by `gdcm::SequenceOfItems::Read()`.

25.69.3.36 `template<typename TDE , typename TSwap > std::ostream const& gdcm::DataSet::Write (std::ostream & os) const`

Reimplemented in `gdcm::FileMetaInformation`, and `gdcm::CommandDataSet`.

25.69.4 Friends And Related Function Documentation

25.69.4.1 `friend class CSAHeader [friend]`

25.69.4.2 `std::ostream& operator<< (std::ostream & _os, const DataSet & val) [friend]`

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

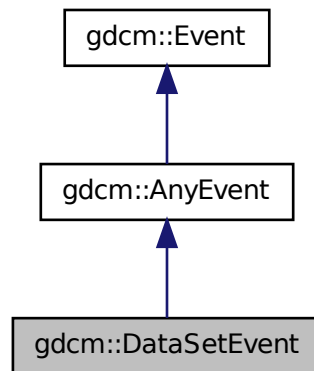
- `gdcmDataSet.h`

25.70 gdcm::DataSetEvent Class Reference

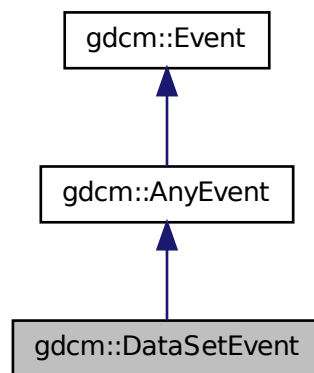
`DataSetEvent` Special type of event triggered during the `DataSet` store/move process.

```
#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

25.70.1 Detailed Description

DataSetEvent Special type of event triggered during the DataSet store/move process.

See also

25.70.2 Member Typedef Documentation

25.70.2.1 `typedef DataSetEvent gdcm::DataSetEvent::Self`

25.70.2.2 `typedef AnyEvent gdcm::DataSetEvent::Superclass`

25.70.3 Constructor & Destructor Documentation

25.70.3.1 `gdcm::DataSetEvent::DataSetEvent (DataSet const * ds = NULL) [inline]`

25.70.3.2 `virtual gdcm::DataSetEvent::~~DataSetEvent () [inline],[virtual]`

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

25.70.4 Member Function Documentation

25.70.4.1 `virtual bool gdcm::DataSetEvent::CheckEvent (const ::gdcm::Event * e) const [inline],[virtual]`

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

25.70.4.3 `virtual const char* gdcm::DataSetEvent::GetEventName () const [inline],[virtual]`

Return the StringName associated with the event.

Implements gdcm::Event.

25.70.4.4 `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

25.71 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)

25.71.1 Detailed Description

DataSetHelper (internal class, not intended for user level)

25.71.2 Member Function Documentation

25.71.2.1 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:

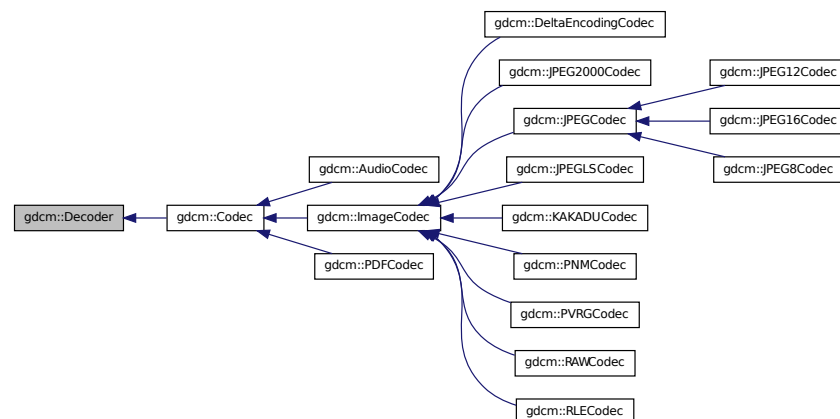
- gdcmDataSetHelper.h

25.72 gdcm::Decoder Class Reference

Decoder.

```
#include <gdcmDecoder.h>
```

Inheritance diagram for gdcm::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 &is_, DataElement &os)`
Decode.

Protected Member Functions

- virtual `bool Decode (std::istream &is_, std::ostream &os)`

25.72.1 Detailed Description

Decoder.

25.72.2 Constructor & Destructor Documentation

25.72.2.1 `virtual gdcm::Decoder::~~Decoder () [inline],[virtual]`

25.72.3 Member Function Documentation

25.72.3.1 `virtual bool gdcm::Decoder::CanDecode (TransferSyntax const &) const [pure virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implemented in `gdcm::JPEGCodec`, `gdcm::PVRGCodec`, `gdcm::RLECodec`, `gdcm::ImageCodec`, `gdcm::JPEG2000Codec`, `gdcm::JPEGLSCodec`, `gdcm::PNMCodec`, `gdcm::RAWCodec`, `gdcm::AudioCodec`, `gdcm::PDFCodec`, and `gdcm::KAKADUCodec`.

25.72.3.2 `virtual bool gdcm::Decoder::Decode (DataElement const &is_, DataElement &os) [inline],[virtual]`

Decode.

Reimplemented in `gdcm::JPEGCodec`, `gdcm::PVRGCodec`, `gdcm::JPEGLSCodec`, `gdcm::JPEG2000Codec`, `gdcm::RLECodec`, `gdcm::ImageCodec`, `gdcm::DeltaEncodingCodec`, `gdcm::KAKADUCodec`, `gdcm::RAWCodec`, `gdcm::AudioCodec`, and `gdcm::PDFCodec`.

25.72.3.3 `virtual bool gdcm::Decoder::Decode (std::istream &is_, std::ostream &os) [inline],[protected],[virtual]`

Reimplemented in `gdcm::JPEGCodec`, `gdcm::JPEG2000Codec`, `gdcm::RLECodec`, `gdcm::ImageCodec`, `gdcm::RAWCodec`, `gdcm::DeltaEncodingCodec`, `gdcm::JPEG12Codec`, `gdcm::JPEG16Codec`, and `gdcm::JPEG8Codec`.

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

- `gdcmDecoder.h`

25.73 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 ()

25.73.1 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.

25.73.2 Constructor & Destructor Documentation

25.73.2.1 gdcm::DefinedTerms::DefinedTerms () [inline]

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

- gdcmDefinedTerms.h

25.74 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`

25.74.1 Detailed Description

FIXME I do not like the name 'Defs'.

Note

bla

Examples:

`GenerateStandardSOPClasses.cxx`, and `TraverseModules.cxx`.

25.74.2 Constructor & Destructor Documentation

25.74.2.1 `gdcm::Defs::Defs ()`

25.74.2.2 `gdcm::Defs::~~Defs ()`

25.74.3 Member Function Documentation

25.74.3.1 `const IOD& gdcm::Defs::GetIODFromFile (const File & file) const`

25.74.3.2 `static const char* gdcm::Defs::GetIODNameFromMediaStorage (MediaStorage const & ms) [static]`

Examples:

`GenerateStandardSOPClasses.cxx`.

25.74.3.3 `const IODs& gdcm::Defs::GetIODs () const [inline]`

25.74.3.4 `IODs& gdcm::Defs::GetIODs () [inline]`

25.74.3.5 `const Macros& gdcm::Defs::GetMacros () const [inline]`

Users should not directly use Macro. Macro are simply a way for DICOM WG to re-use Tables. Macros are convienly wrapped within Modules. See gdcm::Module API directly

25.74.3.6 `Macros& gdcm::Defs::GetMacros () [inline]`

25.74.3.7 `const Modules& gdcm::Defs::GetModules () const [inline]`

25.74.3.8 `Modules& gdcm::Defs::GetModules () [inline]`

25.74.3.9 `Type gdcm::Defs::GetTypeFromTag (const File & file, const Tag & tag) const`

25.74.3.10 `bool gdcm::Defs::IsEmpty () const [inline]`

25.74.3.11 `void gdcm::Defs::LoadDefaults () [protected]`

25.74.3.12 `void gdcm::Defs::LoadFromFile (const char * filename) [protected]`

25.74.3.13 `bool gdcm::Defs::Verify (const File & file) const`

25.74.3.14 `bool gdcm::Defs::Verify (const DataSet & ds) const`

25.74.4 Friends And Related Function Documentation

25.74.4.1 `friend class Global [friend]`

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

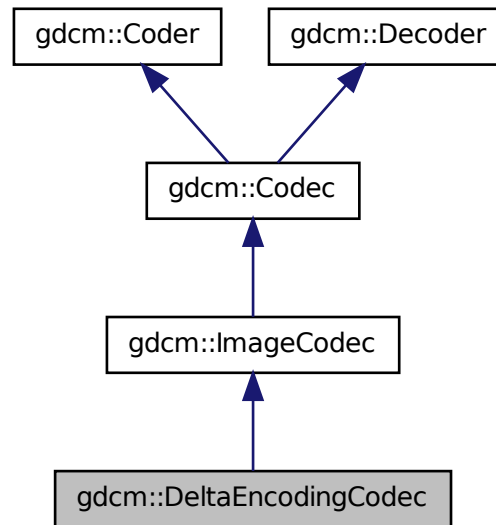
- gdcmDefs.h

25.75 gdcm::DeltaEncodingCodec Class Reference

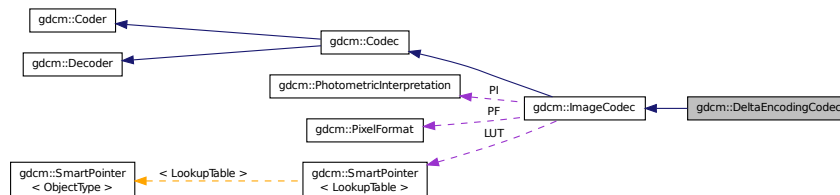
DeltaEncodingCodec compression used by some private vendor.

```
#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

25.75.1 Detailed Description

DeltaEncodingCodec compression used by some private vendor.

25.75.2 Constructor & Destructor Documentation

25.75.2.1 `gdcm::DeltaEncodingCodec::DeltaEncodingCodec ()`

25.75.2.2 `gdcm::DeltaEncodingCodec::~~DeltaEncodingCodec ()`

25.75.3 Member Function Documentation

25.75.3.1 `bool gdcm::DeltaEncodingCodec::CanDecode (TransferSyntax const & ts)`

25.75.3.2 `bool gdcm::DeltaEncodingCodec::Decode (DataElement const & is, DataElement & os)` [virtual]

Decode.

Reimplemented from `gdcm::ImageCodec`.

25.75.3.3 `bool gdcm::DeltaEncodingCodec::Decode (std::istream & is, std::ostream & os)` [protected], [virtual]

Reimplemented from `gdcm::ImageCodec`.

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

- `gdcmDeltaEncodingCodec.h`

25.76 gdcm::DICOMDIR Class Reference

DICOMDIR class.

```
#include <gdcmDICOMDIR.h>
```

Public Member Functions

- `DICOMDIR ()`
- `DICOMDIR (const FileSet &fs)`

25.76.1 Detailed Description

DICOMDIR class.

Structured for handling DICOMDIR

25.76.2 Constructor & Destructor Documentation

25.76.2.1 `gdcm::DICOMDIR::DICOMDIR ()` *[inline]*

25.76.2.2 `gdcm::DICOMDIR::DICOMDIR (const FileSet & fs)` *[inline]*

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

- `gdcmDICOMDIR.h`

25.77 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 ()`

25.77.1 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 gdcm::Scanner does not allow us See PS 3.11 / Table D.3-2 STD-GEN Additional DICOMDIR Keys

25.77.2 Member Typedef Documentation

25.77.2.1 `typedef Directory::FileNamesType gdcm::DICOMDIRGenerator::FileNamesType`

25.77.2.2 `typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType`

25.77.3 Constructor & Destructor Documentation

25.77.3.1 `gdcm::DICOMDIRGenerator::DICOMDIRGenerator ()`

25.77.3.2 `gdcm::DICOMDIRGenerator::~~DICOMDIRGenerator ()`

25.77.4 Member Function Documentation

25.77.4.1 `bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ()` [protected]

25.77.4.2 `bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ()` [protected]

25.77.4.3 `bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ()` [protected]

25.77.4.4 `bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ()` [protected]

25.77.4.5 `bool gdcm::DICOMDIRGenerator::Generate ()`

Main function to generate the DICOMDIR.

25.77.4.6 **File** & gdcmm::DICOMDIRGenerator::GetFile ()

25.77.4.7 **Scanner** & gdcmm::DICOMDIRGenerator::GetScanner () [protected]

25.77.4.8 void gdcmm::DICOMDIRGenerator::SetDescriptor (const char * *d*)

Set the File Set ID.

Warning

this need to be a valid VR::CS value

25.77.4.9 void gdcmm::DICOMDIRGenerator::SetFile (const File & *f*)

Set/Get file. The DICOMDIR file will be valid once a call to Generate has been done.

25.77.4.10 void gdcmm::DICOMDIRGenerator::SetFilenames (FilenamesType const & *fns*)

Set the list of filenames from which the DICOMDIR should be generated from.

25.77.4.11 void gdcmm::DICOMDIRGenerator::SetRootDirectory (FilenameType const & *root*)

Set the root directory from which the filenames should be considered.

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

- gdcmmDICOMDIRGenerator.h

25.78 gdcmm::Dict Class Reference

Class to represent a map of DictEntry.

```
#include <gdcmmDict.h>
```

Public Types

- typedef MapDictEntry::const_iterator ConstIterator
- typedef MapDictEntry::iterator Iterator
- typedef std::map< Tag, DictEntry > MapDictEntry

Public Member Functions

- Dict ()
- 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)

25.78.1 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.

25.78.2 Member Typedef Documentation

25.78.2.1 `typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator`

25.78.2.2 `typedef MapDictEntry::iterator gdcm::Dict::Iterator`

25.78.2.3 `typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry`

25.78.3 Constructor & Destructor Documentation

25.78.3.1 `gdcm::Dict::Dict () [inline]`

25.78.4 Member Function Documentation

25.78.4.1 `void gdcm::Dict::AddDictEntry (const Tag & tag, const DictEntry & de) [inline]`

25.78.4.2 `ConstIterator gdcm::Dict::Begin () const [inline]`

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

25.78.4.3 `ConstIterator gdcmm::Dict::End () const` `[inline]`

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

25.78.4.4 `const DictEntry& gdcmm::Dict::GetDictEntry (const Tag & tag) const` `[inline]`

Examples:

GenFakeIdentifyFile.cxx, and PublicDict.cxx.

25.78.4.5 `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.

25.78.4.6 `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.

25.78.4.7 `const char* gdcmm::Dict::GetKeywordFromTag (Tag const & tag) const` `[inline]`

Function to return the Keyword from a Tag.

25.78.4.8 `bool gdcmm::Dict::IsEmpty () const` `[inline]`

Referenced by gdcmm::Dicts::IsEmpty().

25.78.4.9 `void gdcmm::Dict::LoadDefault ()` `[protected]`

25.78.5 Friends And Related Function Documentation

25.78.5.1 `friend class Dicts` `[friend]`

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

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

- gdcmmDict.h

25.79 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 ()

25.79.1 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

25.79.2 Member Enumeration Documentation

25.79.2.1 enum gdcm::DictConverter::OutputTypes

Enumerator:

DICT_DEFAULT

DICT_DEBUG

DICT_XML

25.79.3 Constructor & Destructor Documentation

25.79.3.1 gdcm::DictConverter::DictConverter ()

25.79.3.2 gdcm::DictConverter::~~DictConverter ()

25.79.4 Member Function Documentation

25.79.4.1 void gdcm::DictConverter::AddGroupLength () [protected]

25.79.4.2 void gdcm::DictConverter::Convert ()

25.79.4.3 bool gdcm::DictConverter::ConvertToCXX (const char * raw, std::string & cxx) [protected]

25.79.4.4 bool gdcm::DictConverter::ConvertToXML (const char * raw, std::string & cxx) [protected]

25.79.4.5 const std::string& gdcm::DictConverter::GetDictName () const

25.79.4.6 const std::string& gdcm::DictConverter::GetInputFilename () const

25.79.4.7 const std::string& gdcm::DictConverter::GetOutputFilename () const

25.79.4.8 int gdcm::DictConverter::GetOutputType () const [inline]

25.79.4.9 static bool gdcm::DictConverter::Readuint16 (const char * raw, uint16_t & ov) [static]

25.79.4.10 static bool gdcm::DictConverter::ReadVM (const char * raw, VM::VMType & type) [static]

25.79.4.11 static bool gdcm::DictConverter::ReadVR (const char * raw, VR::VRType & type) [static]

25.79.4.12 void gdcm::DictConverter::SetDictName (const char * name)

25.79.4.13 void gdcm::DictConverter::SetInputFileName (const char * filename)

25.79.4.14 void gdcm::DictConverter::SetOutputFileName (const char * filename)

25.79.4.15 void gdcm::DictConverter::SetOutputType (int type) [inline]

25.79.4.16 void gdcmmDictConverter::WriteFooter () [protected]

25.79.4.17 void gdcmmDictConverter::WriteHeader () [protected]

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

- gdcmmDictConverter.h

25.80 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

- std::ostream & operator<< (std::ostream &_os, const DictEntry &_val)

25.80.1 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`.

25.80.2 Constructor & Destructor Documentation

25.80.2.1 `gdcm::DictEntry::DictEntry (const char * name = " ", const char * keyword = " ", VR const & vr = VR::INVALID, VM const & vm = VM::VMO, bool ret = false) [inline]`

25.80.3 Member Function Documentation

25.80.3.1 `const char* gdcm::DictEntry::GetKeyword () const [inline]`

same as `GetName` but without spaces...

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

Set/Get Name.

Referenced by `gdcm::PrivateDict::PrintXML()`.

25.80.3.3 `bool gdcm::DictEntry::GetRetired () const [inline]`

Set/Get Retired flag.

Examples:

`GenAllVR.cxx`.

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

Set/Get VM.

Referenced by `gdcm::PrivateDict::AddDictEntry()`, and `gdcm::PrivateDict::PrintXML()`.

25.80.3.5 `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()`.

25.80.3.6 `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)

25.80.3.7 `void gdcm::DictEntry::SetElementXX (bool v)` `[inline]`

Set whether element is shared in multiple elements (Source Image IDs typically)

25.80.3.8 `void gdcm::DictEntry::SetGroupXX (bool v)` `[inline]`

Set whether element is shared in multiple groups (Curve/Overlay typically)

25.80.3.9 `void gdcm::DictEntry::SetKeyword (const char * keyword)` `[inline]`

25.80.3.10 `void gdcm::DictEntry::SetName (const char * name)` `[inline]`

25.80.3.11 `void gdcm::DictEntry::SetRetired (bool retired)` `[inline]`

25.80.3.12 `void gdcm::DictEntry::SetVM (VM const & vm)` `[inline]`

25.80.3.13 `void gdcm::DictEntry::SetVR (const VR & vr)` `[inline]`

Referenced by `gdcm::PrivateDict::AddDictEntry()`.

25.80.4 Friends And Related Function Documentation

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

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

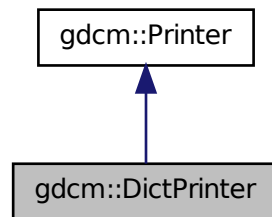
- `gdcmDictEntry.h`

25.81 gdcm::DictPrinter Class Reference

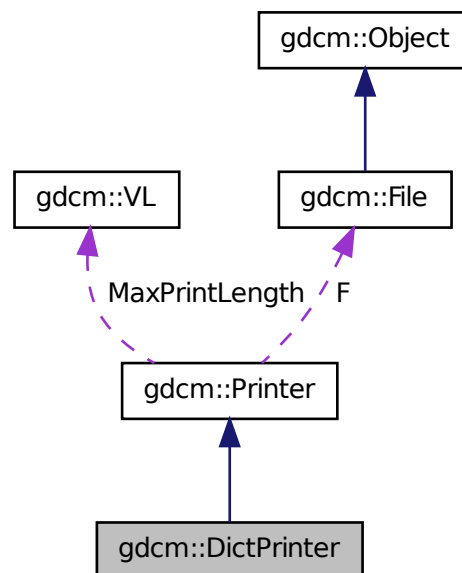
DictPrinter class.

```
#include <gdcmDictPrinter.h>
```

Inheritance diagram for `gdcM::DictPrinter`:



Collaboration diagram for `gdcM::DictPrinter`:



Public Member Functions

- `DictPrinter ()`
- `~DictPrinter ()`
- `void Print (std::ostream &os)`

Print.

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

25.81.1 Detailed Description

DictPrinter class.

25.81.2 Constructor & Destructor Documentation

25.81.2.1 gdcm::DictPrinter::DictPrinter ()

25.81.2.2 gdcm::DictPrinter::~~DictPrinter ()

25.81.3 Member Function Documentation

25.81.3.1 void gdcm::DictPrinter::Print (std::ostream & os)

Print.

Reimplemented from gdcm::Printer.

25.81.3.2 void gdcm::DictPrinter::PrintDataElement2 (std::ostream & os, const DataSet & ds, const DataElement & ide)
[protected]

25.81.3.3 void gdcm::DictPrinter::PrintDataSet2 (std::ostream & os, const DataSet & ds) [protected]

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

- gdcmDictPrinter.h

25.82 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
- 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)

25.82.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load)

Note

bla

Examples:

GenAllVR.cxx, GenFakeldentifyFile.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

25.82.2 Member Enumeration Documentation

25.82.2.1 enum gdcmm::Dicts::ConstructorType [protected]

Enumerator:

PHILIPS

GEMS

SIEMENS

25.82.3 Constructor & Destructor Documentation

25.82.3.1 gdcmm::Dicts::Dicts ()

25.82.3.2 gdcmm::Dicts::~~Dicts ()

25.82.4 Member Function Documentation

25.82.4.1 static const char* gdcmm::Dicts::GetConstructorString (**ConstructorType** *type*) [static], [protected]

25.82.4.2 const CSAHeaderDict& gdcmm::Dicts::GetCSAHeaderDict () const

Examples:

MrProtocol.cxx.

25.82.4.3 const DictEntry& gdcmm::Dicts::GetDictEntry (const Tag & *tag*, const char * *owner* = NULL) const

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.

25.82.4.4 const DictEntry& gdcmm::Dicts::GetDictEntry (const PrivateTag & *tag*) const

25.82.4.5 const PrivateDict& gdcmm::Dicts::GetPrivateDict () const

25.82.4.6 PrivateDict& gdcmm::Dicts::GetPrivateDict ()

25.82.4.7 const Dict& gdcmm::Dicts::GetPublicDict () const

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

25.82.4.8 bool gdcmm::Dicts::IsEmpty () const [inline]

References gdcmm::Dict::IsEmpty().

25.82.4.9 void gdcmm::Dicts::LoadDefaults () [protected]

25.82.5 Friends And Related Function Documentation

25.82.5.1 friend class Global [friend]

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

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

- `gdcmDicts.h`

25.83 `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 }`

25.83.1 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)

25.83.2 Member Enumeration Documentation

25.83.2.1 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:

- gdcmDIMSE.h

25.84 gdcm::DirectionCosines Class Reference

class to handle DirectionCosines

```
#include <gdcmDirectionCosines.h>
```

Public Member Functions

- DirectionCosines ()
- DirectionCosines (const double dircos[6])
- ~DirectionCosines ()
- double ComputeDistAlongNormal (const double ipp[3]) const
Compute the distance along the normal.

- void Cross (double z[3]) const
Compute Cross product.
- double CrossDot (DirectionCosines const &dc) const
Compute the Dot product of the two cross vector of both DirectionCosines object.
- double Dot () const
Compute Dot.
- bool IsValid () const
Return whether or not this is a valid direction cosines.
- void Normalize ()
Normalize in-place.
- operator const double * () const
*Make the class behave like a const double *.*
- void Print (std::ostream &) const
Print.
- bool SetFromString (const char *str)

25.84.1 Detailed Description

class to handle DirectionCosines

Examples:

DiscriminateVolume.cxx.

25.84.2 Constructor & Destructor Documentation

25.84.2.1 `gdc::DirectionCosines::DirectionCosines ()`

25.84.2.2 `gdc::DirectionCosines::DirectionCosines (const double dircos[6])`

25.84.2.3 `gdc::DirectionCosines::~~DirectionCosines ()`

25.84.3 Member Function Documentation

25.84.3.1 `double gdc::DirectionCosines::ComputeDistAlongNormal (const double ipp[3]) const`

Compute the distance along the normal.

25.84.3.2 `void gdc::DirectionCosines::Cross (double z[3]) const`

Compute Cross product.

25.84.3.3 `double gdc::DirectionCosines::CrossDot (DirectionCosines const & dc) const`

Compute the Dot product of the two cross vector of both DirectionCosines object.

Examples:

DiscriminateVolume.cxx.

25.84.3.4 `double gdcm::DirectionCosines::Dot () const`

Compute Dot.

25.84.3.5 `bool gdcm::DirectionCosines::IsValid () const`

Return whether or not this is a valid direction cosines.

25.84.3.6 `void gdcm::DirectionCosines::Normalize ()`

Normalize in-place.

25.84.3.7 `gdcm::DirectionCosines::operator const double * () const` `[inline]`

Make the class behave like a const double *.

25.84.3.8 `void gdcm::DirectionCosines::Print (std::ostream &) const`

Print.

25.84.3.9 `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

25.85 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)

25.85.1 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 gdcM::Directory does not make much sense.

Examples:

DiscriminateVolume.cxx, DumpToSQLite3.cxx, gdcMorthoplanes.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcM.cxx, and VolumeSorter.cxx.

25.85.2 Member Typedef Documentation

25.85.2.1 typedef std::vector<FilenameType> gdcM::Directory::FileNamesType

Examples:

DiscriminateVolume.cxx.

25.85.2.2 `typedef std::string gdcmm::Directory::FilenameType`

25.85.3 Constructor & Destructor Documentation

25.85.3.1 `gdcmm::Directory::Directory ()` `[inline]`

25.85.3.2 `gdcmm::Directory::~~Directory ()` `[inline]`

25.85.4 Member Function Documentation

25.85.4.1 `unsigned int gdcmm::Directory::Explore (FilenameType const & name, bool recursive)` `[protected]`

Return number of file found when 'recursive'ly exploring directory name

25.85.4.2 `FilenameType const& gdcmm::Directory::GetDirectories () const` `[inline]`

Return the Directories traversed.

25.85.4.3 `FilenameType const& gdcmm::Directory::GetFilenames () const` `[inline]`

Set/Get the file names within the directory.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmmorthoplanes.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcmm.cxx, and VolumeSorter.cxx.

25.85.4.4 `FilenameType const& gdcmm::Directory::GetToplevel () const` `[inline]`

Get the name of the toplevel directory.

25.85.4.5 `unsigned int gdcmm::Directory::Load (FilenameType const & name, bool recursive = false)` `[inline]`

construct a list of filenames and subdirectory beneath directory: name

Warning

: hidden file and hidden directory are not loaded.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmmorthoplanes.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcmm.cxx, and VolumeSorter.cxx.

25.85.4.6 `void gdcmm::Directory::Print (std::ostream & os = std::cout) const`

Print.

Examples:

SortImage.cxx.

Referenced by `gdcmm::operator<<()`.

25.85.5 Friends And Related Function Documentation

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

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

- `gdcmmDirectory.h`

25.86 gdcmm::DirectoryHelper Class Reference

```
#include <gdcmmDirectoryHelper.h>
```

Static Public Member Functions

- static `Directory::FilenameType GetCTImageSeriesUIDs (const std::string &inDirectory)`
- static `Directory::FilenameType GetFileNamesFromSeriesUIDs (const std::string &inDirectory, const std::string &inSeriesUID)`
- static `std::string GetFrameOfReference (const std::vector< DataSet > &inDS)`
- static `Directory::FilenameType GetMRImageSeriesUIDs (const std::string &inDirectory)`
- static `Directory::FilenameType GetRTStructSeriesUIDs (const std::string &inDirectory)`
- static `Directory::FilenameType GetSeriesUIDsBySOPClassUID (const std::string &inDirectory, const std::string &inSOPClassUID)`
- static `std::string GetSOPClassUID (const std::vector< DataSet > &inDS)`
- static `std::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)`

25.86.1 Member Function Documentation

25.86.1.1 `static Directory::FilenameType gdcmm::DirectoryHelper::GetCTImageSeriesUIDs (const std::string & inDirectory)` [static]

25.86.1.2 `static Directory::FilenameType gdcmm::DirectoryHelper::GetFileNamesFromSeriesUIDs (const std::string & inDirectory, const std::string & inSeriesUID)` [static]

Examples:

GenerateRTSTRUCT.cxx.

25.86.1.3 static std::string gdcm::DirectoryHelper::GetFrameOfReference (const std::vector< DataSet > & *inDS*) [static]

25.86.1.4 static Directory::FileNamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (const std::string & *inDirectory*) [static]

25.86.1.5 static Directory::FileNamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (const std::string & *inDirectory*) [static]

Examples:

GenerateRTSTRUCT.cxx.

25.86.1.6 static Directory::FileNamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (const std::string & *inDirectory*, const std::string & *inSOPClassUID*) [static]

25.86.1.7 static std::string gdcm::DirectoryHelper::GetSOPClassUID (const std::vector< DataSet > & *inDS*) [static]

25.86.1.8 static std::vector<DataSet> gdcm::DirectoryHelper::LoadImageFromFiles (const std::string & *inDirectory*, const std::string & *inSeriesUID*) [static]

25.86.1.9 static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (int *inIndex*, const std::vector< DataSet > & *inDS*) [static]

25.86.1.10 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

25.87 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcmDummyValueGenerator.h>
```

Static Public Member Functions

- static const char * Generate (const char *input)

25.87.1 Detailed Description

Class for generating dummy value.

See also

Anonymizer

25.87.2 Member Function Documentation

25.87.2.1 `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:

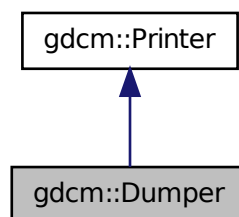
- `gdcmDummyValueGenerator.h`

25.88 `gdcm::Dumper` Class Reference

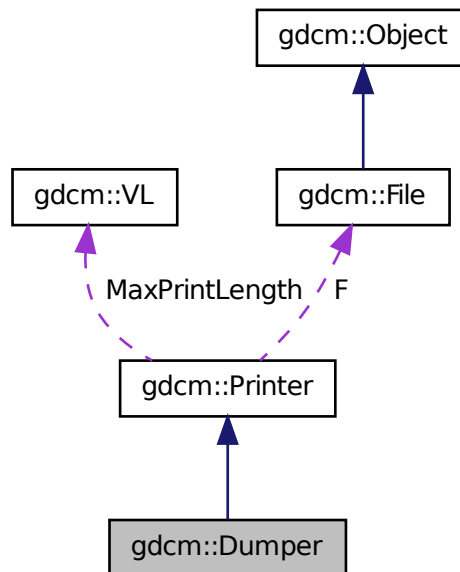
Codec class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for `gdcm::Dumper`:



Collaboration diagram for gdcm::Dumper:



Public Member Functions

- `Dumper ()`
- `~Dumper ()`

Additional Inherited Members

25.88.1 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

25.88.2 Constructor & Destructor Documentation

25.88.2.1 `gdcm::Dumper::Dumper ()` `[inline]`

25.88.2.2 `gdcm::Dumper::~Dumper ()` `[inline]`

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

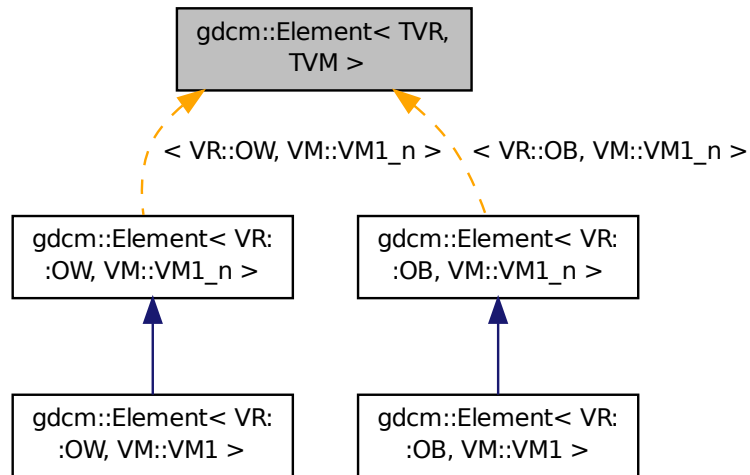
- `gdcmDumper.h`

25.89 gdcm::Element< TVR, TVM > Class Template Reference

Element class.

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< 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)

25.89.1 Detailed Description

template<int TVR, int TVM>class gdcmm::Element< TVR, TVM >

Element class.

Note

TODO

Examples:

csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_ - Stream_Image_Writer.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

25.89.2 Member Typedef Documentation

25.89.2.1 template<int TVR, int TVM> typedef VRToType<TVR>::Type gdcmm::Element< TVR, TVM >::Type

25.89.3 Member Function Documentation

25.89.3.1 template<int TVR, int TVM> DataElement gdcmm::Element< TVR, TVM >::GetAsDataElement () const
[inline]

25.89.3.2 template<int TVR, int TVM> unsigned long gdcmm::Element< TVR, TVM >::GetLength () const [inline]

Referenced by gdcmm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcmm::Element< TVR, VM::VM1_n >::GetAsDataElement(), gdcmm::Element< TVR, VM::VM1_n >::Print(), gdcmm::Element< VR::OB, VM::VM1_n >::Read(), gdcmm::Element< TVR, VM::VM1_n >::Read(), gdcmm::Element< VR::OB, VM::VM1_n >::Set(), gdcmm::Element< TVR, VM::VM1_n >::Set(), gdcmm::Element< VR::OB, VM::VM1_n >::SetNoSwap(), gdcmm::Element< TVR, VM::VM1_n >::SetNoSwap(), gdcmm::Element< VR::OB, VM::VM1_n >::Write(), gdcmm::Element< TVR, VM::VM1_n >::Write(), and gdcmm::Element< TVR, VM::VM1_n >::WriteASCII().

25.89.3.3 template<int TVR, int TVM> const VRToType<TVR>::Type& gdcmm::Element< TVR, TVM >::GetValue (unsigned int idx = 0) const [inline]

Referenced by gdcmm::Element< VR::OB, VM::VM1_n >::operator[](), and gdcmm::Element< TVR, VM::VM1_n >::operator[]().

25.89.3.4 `template<int TVR, int TVM> VRToType<TVR>::Type& gdcM::Element< TVR, TVM >::GetValue (unsigned int idx = 0) [inline]`

25.89.3.5 `template<int TVR, int TVM> const VRToType<TVR>::Type* gdcM::Element< TVR, TVM >::GetValues () const [inline]`

25.89.3.6 `template<int TVR, int TVM> static VM gdcM::Element< TVR, TVM >::GetVM () [inline],[static]`

25.89.3.7 `template<int TVR, int TVM> static VR gdcM::Element< TVR, TVM >::GetVR () [inline],[static]`

Referenced by `gdcM::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, and `gdcM::Element< TVR, VM::VM1_n >::GetAsDataElement()`.

25.89.3.8 `template<int TVR, int TVM> VRToType<TVR>::Type gdcM::Element< TVR, TVM >::operator[] (unsigned int idx) const [inline]`

25.89.3.9 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Print (std::ostream & _os) const [inline]`

25.89.3.10 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Read (std::istream & _is) [inline]`

Referenced by `gdcM::Element< VR::OB, VM::VM1_n >::Read()`, `gdcM::Element< TVR, VM::VM1_n >::Read()`, `gdcM::Element< VR::OB, VM::VM1_n >::Set()`, and `gdcM::Element< TVR, VM::VM1_n >::Set()`.

25.89.3.11 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Set (Value const & v) [inline]`

Referenced by `gdcM::Element< VR::OB, VM::VM1_n >::SetFromDataElement()`, and `gdcM::Element< TVR, VM::VM1_n >::SetFromDataElement()`.

25.89.3.12 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetFromDataElement (DataElement< TVR, TVM > const & de) [inline]`

25.89.3.13 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetNoSwap (Value const & v) [inline],[protected]`

Referenced by `gdcM::Element< VR::OB, VM::VM1_n >::SetFromDataElement()`, and `gdcM::Element< TVR, VM::VM1_n >::SetFromDataElement()`.

25.89.3.14 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetValue (typename VRToType< TVR >::Type v, unsigned int idx = 0) [inline]`

25.89.3.15 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Write (std::ostream & _os) const [inline]`

Referenced by `gdcM::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, `gdcM::Element< TVR, VM::VM1_n >::GetAsDataElement()`, `gdcM::Element< VR::OB, VM::VM1_n >::Write()`, `gdcM::Element< TVR, VM::VM1_n >::Write()`, and `gdcM::Element< TVR, VM::VM1_n >::WriteASCII()`.

25.89.4 Member Data Documentation

25.89.4.1 `template<int TVR, int TVM> VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]`

Referenced by `gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, `gdcm::Element< VR::OB, VM::VM1_n >::GetValue()`, `gdcm::Element< VR::OB, VM::VM1_n >::GetValues()`, `gdcm::Element< VR::OB, VM::VM1_n >::Print()`, `gdcm::Element< VR::AS, VM::VM5 >::Print()`, `gdcm::Element< VR::OB, VM::VM1_n >::Read()`, `gdcm::Element< VR::OB, VM::VM1_n >::Set()`, `gdcm::Element< TVR, VM::VM1_n >::SetLength()`, `gdcm::Element< VR::OB, VM::VM1_n >::SetNoSwap()`, `gdcm::Element< VR::OB, VM::VM1_n >::SetValue()`, `gdcm::Element< VR::OB, VM::VM1_n >::Write()`, and `gdcm::Element< TVR, VM::VM1_n >::~~Element()`.

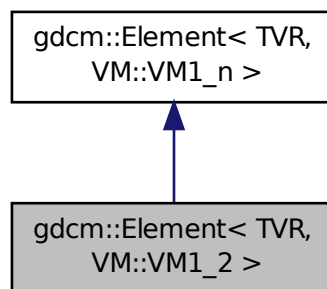
The documentation for this class was generated from the following file:

- `gdcmElement.h`

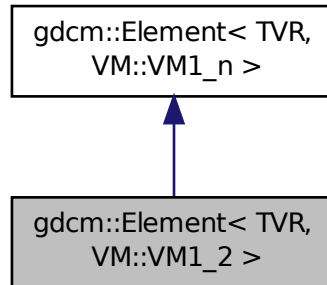
25.90 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference

```
#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

25.90.1 Member Typedef Documentation

25.90.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM1_2 >::Parent`

25.90.2 Member Function Documentation

25.90.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM1_2 >::SetLength (int len) [inline]`

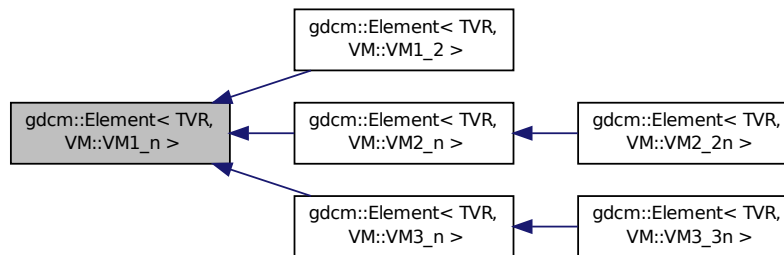
The documentation for this class was generated from the following file:

- `gdcmElement.h`

25.91 `gdcm::Element< TVR, VM::VM1_n >` Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM1_n >:



Public Types

- typedef `VRToType< TVR >::Type` Type

Public Member Functions

- `Element ()`
- `Element (const Element &_val)`
- `~Element ()`
- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`
- `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)`

25.91.1 Member Typedef Documentation

25.91.1.1 `template<int TVR> typedef VRToType<TVR>::Type gdcmm::Element< TVR, VM::VM1_n >::Type`

25.91.2 Constructor & Destructor Documentation

25.91.2.1 `template<int TVR> gdcmm::Element< TVR, VM::VM1_n >::Element () [inline],[explicit]`

25.91.2.2 `template<int TVR> gdcmm::Element< TVR, VM::VM1_n >::~~Element () [inline]`

References `gdcmm::Element< TVR, TVM >::Internal`.

25.91.2.3 `template<int TVR> gdcmm::Element< TVR, VM::VM1_n >::Element (const Element< TVR, VM::VM1_n > &_val) [inline]`

25.91.3 Member Function Documentation

25.91.3.1 `template<int TVR> DataElement gdcmm::Element< TVR, VM::VM1_n >::GetAsDataElement () const [inline]`

References `gdcmm::Element< TVR, TVM >::GetLength()`, `gdcmm::Element< TVR, TVM >::GetVR()`, `gdcmm::DataElement::GetVR()`, `gdcmm::DataElement::SetByteValue()`, `gdcmm::DataElement::SetVR()`, `gdcmm::VR::SQ`, `gdcmm::VR::UI`, `gdcmm::VR::VRASCII`, and `gdcmm::Element< TVR, TVM >::Write()`.

25.91.3.2 `template<int TVR> unsigned long gdcmm::Element< TVR, VM::VM1_n >::GetLength () const [inline]`

25.91.3.3 `template<int TVR> const VRToType<TVR>::Type& gdcmm::Element< TVR, VM::VM1_n >::GetValue (unsigned int idx = 0) const [inline]`

25.91.3.4 `template<int TVR> VRToType<TVR>::Type& gdcmm::Element< TVR, VM::VM1_n >::GetValue (unsigned int idx = 0) [inline]`

25.91.3.5 `template<int TVR> static VM gdcmm::Element< TVR, VM::VM1_n >::GetVM () [inline],[static]`

References `gdcmm::VM::VM1_n`.

25.91.3.6 `template<int TVR> static VR gdcmm::Element< TVR, VM::VM1_n >::GetVR () [inline],[static]`

25.91.3.7 `template<int TVR> Element& gdcmm::Element< TVR, VM::VM1_n >::operator= (const Element< TVR, VM::VM1_n > &_val) [inline]`

25.91.3.8 `template<int TVR> VRToType<TVR>::Type gdcmm::Element< TVR, VM::VM1_n >::operator[] (unsigned int idx) const [inline]`

References `gdcmm::Element< TVR, TVM >::GetValue()`.

25.91.3.9 `template<int TVR> void gdcmm::Element< TVR, VM::VM1_n >::Print (std::ostream &_os) const [inline]`

References `gdcmm::Element< TVR, TVM >::GetLength()`.

25.91.3.10 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Read (std::istream & is) [inline]`

References `gdcm::Element< TVR, TVM >::GetLength()`, and `gdcm::Element< TVR, TVM >::Read()`.

25.91.3.11 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Set (Value const & v) [inline]`

References `gdcm::Element< TVR, TVM >::GetLength()`, `gdcm::ByteValue::GetLength()`, `gdcm::ByteValue::GetPointer()`, `gdcm::Element< TVR, TVM >::Read()`, and `gdcm::VR::VRBINARY`.

25.91.3.12 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetArray (const Type * array, unsigned long len, bool save = false) [inline]`

25.91.3.13 `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()`, `gdcm::DataElement::GetVR()`, `gdcm::VR::INVALID`, `gdcm::Element< TVR, TVM >::Set()`, `gdcm::Element< TVR, TVM >::SetNoSwap()`, and `gdcm::VR::UN`.

25.91.3.14 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetLength (unsigned long len) [inline]`

References `gdcm::Element< TVR, TVM >::Internal`.

25.91.3.15 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (Value const & v) [inline], [protected]`

References `gdcm::Element< TVR, TVM >::GetLength()`, `gdcm::ByteValue::GetLength()`, `gdcm::ByteValue::GetPointer()`, and `gdcm::VR::VRBINARY`.

25.91.3.16 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetValue (typename VRToType< TVR >::Type v, unsigned int idx = 0) [inline]`

25.91.3.17 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Write (std::ostream & os) const [inline]`

References `gdcm::Element< TVR, TVM >::GetLength()`, and `gdcm::Element< TVR, TVM >::Write()`.

25.91.3.18 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (std::ostream & os) const [inline]`

References `gdcm::Element< TVR, TVM >::GetLength()`, and `gdcm::Element< TVR, TVM >::Write()`.

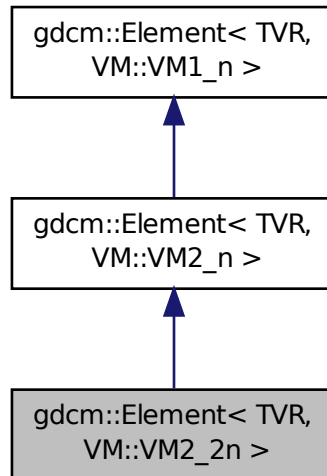
The documentation for this class was generated from the following file:

- `gdcmElement.h`

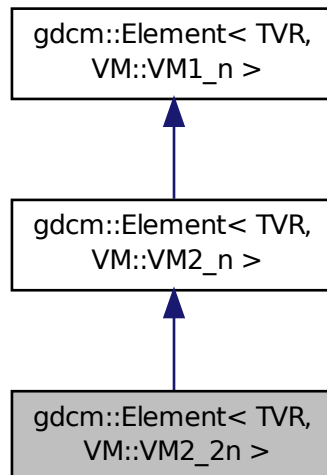
25.92 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference

```
#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)

25.92.1 Member Typedef Documentation

25.92.1.1 `template<int TVR> typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent`

Reimplemented from `gdcm::Element< TVR, VM::VM2_n >`.

25.92.2 Member Function Documentation

25.92.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM2_2n >::SetLength (int len) [inline]`

Reimplemented from `gdcm::Element< TVR, VM::VM2_n >`.

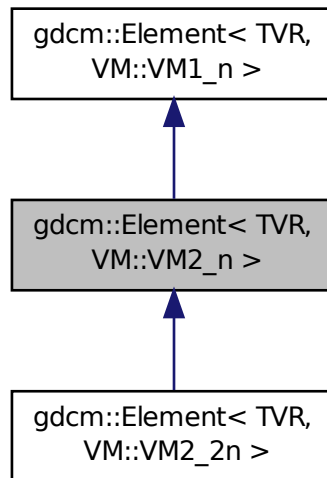
The documentation for this class was generated from the following file:

- `gdcmElement.h`

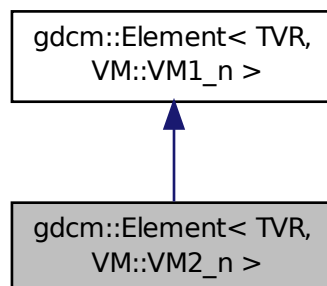
25.93 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< TVR, VM::VM2_n >`:



Collaboration diagram for `gdcm::Element< TVR, VM::VM2_n >`:



Public Types

- `typedef Element< TVR, VM::VM1_n > Parent`

Public Member Functions

- `void SetLength (int len)`

Additional Inherited Members

25.93.1 Member Typedef Documentation

25.93.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcmm::Element< TVR, VM::VM2_n >::Parent`

Reimplemented in `gdcmm::Element< TVR, VM::VM2_2n >`.

25.93.2 Member Function Documentation

25.93.2.1 `template<int TVR> void gdcmm::Element< TVR, VM::VM2_n >::SetLength (int len) [inline]`

Reimplemented in `gdcmm::Element< TVR, VM::VM2_2n >`.

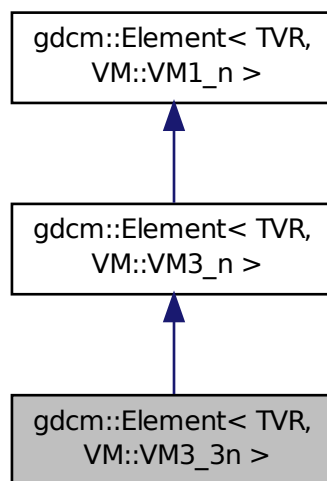
The documentation for this class was generated from the following file:

- `gdcmmElement.h`

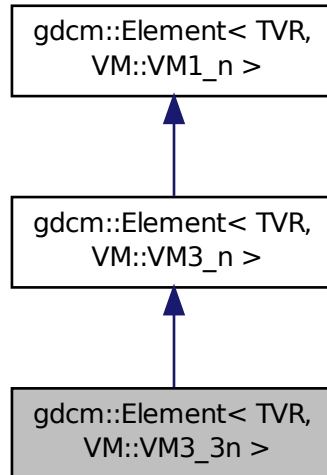
25.94 gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for `gdcmm::Element< TVR, VM::VM3_3n >`:



Collaboration diagram for `gdcM::Element< TVR, VM::VM3_3n >`:



Public Types

- `typedef Element< TVR, VM::VM3_n > Parent`

Public Member Functions

- `void SetLength (int len)`

25.94.1 Member Typedef Documentation

25.94.1.1 `template<int TVR> typedef Element<TVR, VM::VM3_n> gdcM::Element< TVR, VM::VM3_3n >::Parent`

Reimplemented from `gdcM::Element< TVR, VM::VM3_n >`.

25.94.2 Member Function Documentation

25.94.2.1 `template<int TVR> void gdcM::Element< TVR, VM::VM3_3n >::SetLength (int len) [inline]`

Reimplemented from `gdcM::Element< TVR, VM::VM3_n >`.

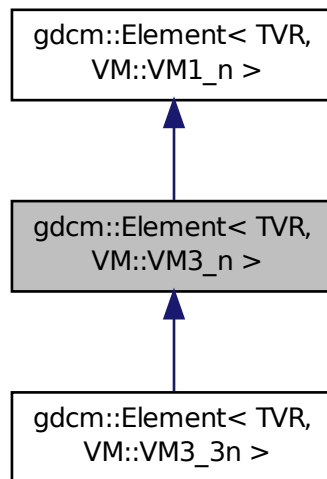
The documentation for this class was generated from the following file:

- `gdcMElement.h`

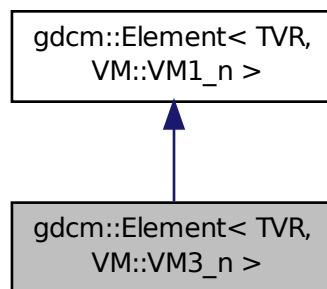
25.95 gdcM::Element< TVR, VM::VM3_n > Class Template Reference

```
#include <gdcMElement.h>
```

Inheritance diagram for gdcM::Element< TVR, VM::VM3_n >:



Collaboration diagram for gdcM::Element< TVR, VM::VM3_n >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

25.95.1 Member Typedef Documentation

25.95.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcmm::Element< TVR, VM::VM3_n >::Parent`

Reimplemented in `gdcmm::Element< TVR, VM::VM3_3n >`.

25.95.2 Member Function Documentation

25.95.2.1 `template<int TVR> void gdcmm::Element< TVR, VM::VM3_n >::SetLength (int len) [inline]`

Reimplemented in `gdcmm::Element< TVR, VM::VM3_3n >`.

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

- `gdcmmElement.h`

25.96 gdcmm::Element< VR::AS, VM::VM5 > Class Template Reference

```
#include <gdcmmElement.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)]

25.96.1 Member Function Documentation

25.96.1.1 `unsigned long gdcmm::Element< VR::AS, VM::VM5 >::GetLength () const [inline]`

25.96.1.2 `void gdcmm::Element< VR::AS, VM::VM5 >::Print (std::ostream &_os) const [inline]`

References `gdcmm::Element< TVR, TVM >::Internal`.

25.96.2 Member Data Documentation

25.96.2.1 `char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]`

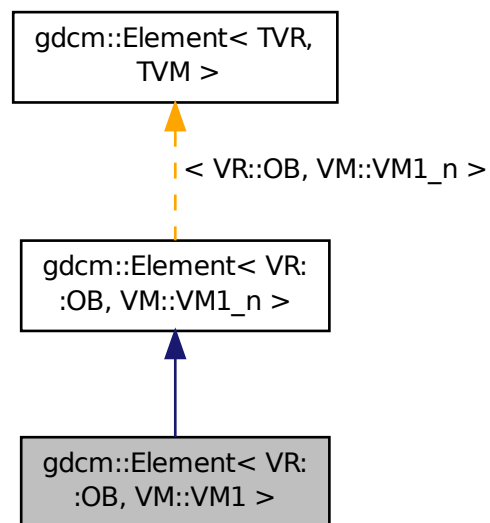
The documentation for this class was generated from the following file:

- gdcmElement.h

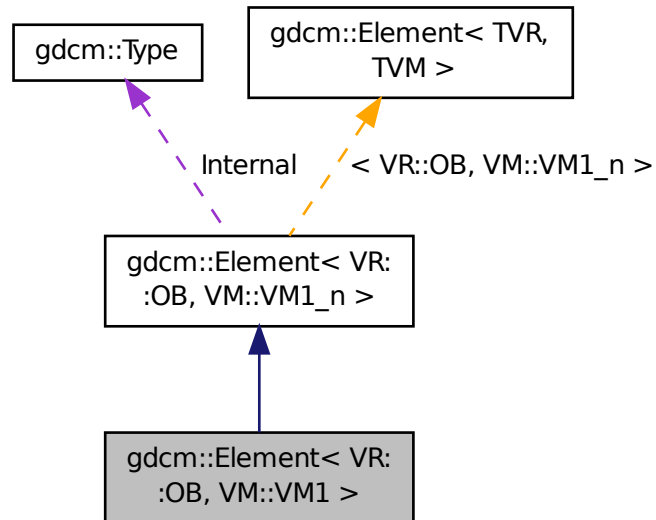
25.97 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for `gdcm::Element< VR::OB, VM::VM1 >`:



Additional Inherited Members

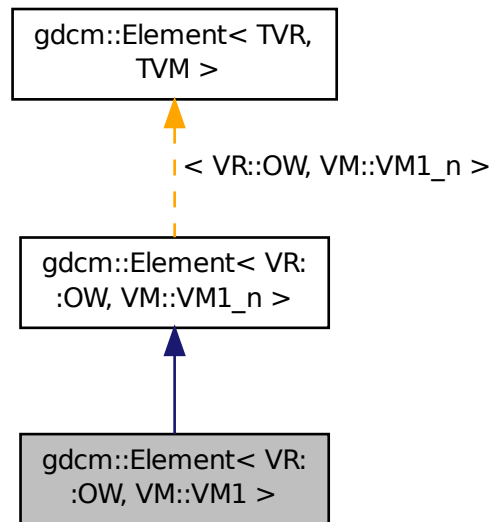
The documentation for this class was generated from the following file:

- `gdcmElement.h`

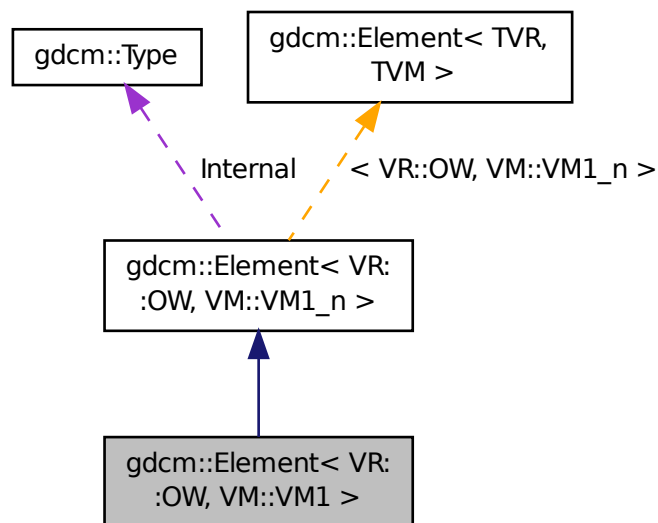
25.98 `gdcm::Element< VR::OW, VM::VM1 >` Class Template Reference

```
#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`

25.99 gdcm::EncapsulatedDocument Class Reference

EncapsulatedDocument.

```
#include <gdcmEncapsulatedDocument.h>
```

Public Member Functions

- `EncapsulatedDocument ()`

25.99.1 Detailed Description

EncapsulatedDocument.

25.99.2 Constructor & Destructor Documentation

25.99.2.1 `gdcm::EncapsulatedDocument::EncapsulatedDocument () [inline]`

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

- `gdcmEncapsulatedDocument.h`

25.100 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)`

25.100.1 Member Function Documentation

25.100.1.1 `template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Read (T * data, unsigned long length, std::istream &_is)` `[inline]`, `[static]`

25.100.1.2 `template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (T * data, unsigned int &length, std::istream &_is)` `[inline]`, `[static]`

References `gdcm::backslash()`.

25.100.1.3 `template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (T * data, unsigned long length, std::istream &_is)` `[inline]`, `[static]`

25.100.1.4 `template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Write (const T * data, unsigned long length, std::ostream &_os)` `[inline]`, `[static]`

25.100.1.5 `template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write (const float * data, unsigned long length, std::ostream &_os)` `[inline]`

References `gdcm::to_string()`.

25.100.1.6 `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`

25.101 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)`

25.101.1 Member Function Documentation

25.101.1.1 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (T * data, unsigned long length, std::istream & is)` `[inline]`, `[static]`

References `gdcm::SwapperNoOp::SwapArray()`.

25.101.1.2 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (T * data, unsigned int & length, std::istream & is)` `[inline]`, `[static]`

25.101.1.3 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (T * data, unsigned long length, std::istream & is)` `[inline]`, `[static]`

25.101.1.4 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (const T * data, unsigned long length, std::ostream & os)` `[inline]`, `[static]`

References `gdcm::SwapperNoOp::Swap()`.

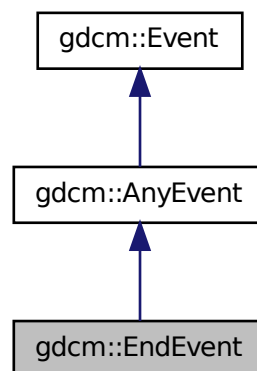
The documentation for this class was generated from the following file:

- `gdcmElement.h`

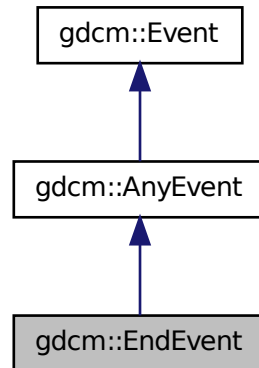
25.102 gdcm::EndEvent Class Reference

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

Inheritance diagram for `gdcm::EndEvent`:



Collaboration diagram for gdcmm::EndEvent:



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

- gdcmmEvent.h

25.103 gdcmm::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 <gdcmmEnumeratedValues.h>
```

Public Member Functions

- EnumeratedValues ()

25.103.1 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.

25.103.2 Constructor & Destructor Documentation

25.103.2.1 `gdcm::EnumeratedValues::EnumeratedValues ()` `[inline]`

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

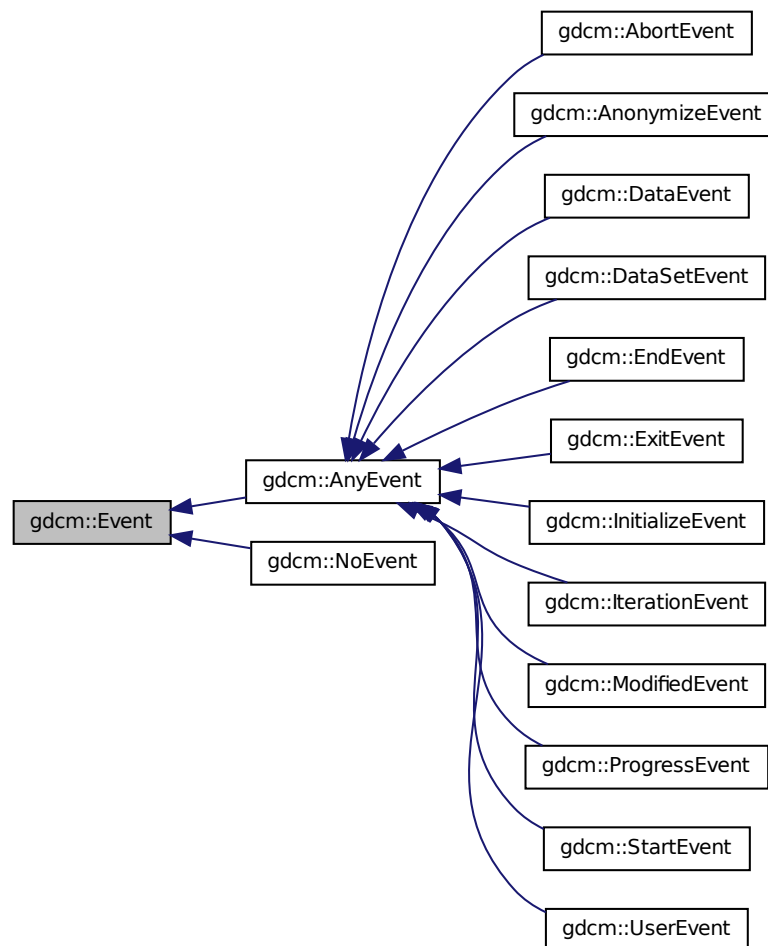
- `gdcmEnumeratedValues.h`

25.104 `gdcm::Event` Class Reference

superclass for callback/observer methods

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

Inheritance diagram for `gdcm::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`

25.104.1 Detailed Description

superclass for callback/observer methods

See also

Command Subject

25.104.2 Constructor & Destructor Documentation

25.104.2.1 `gdcm::Event::Event ()`

25.104.2.2 `gdcm::Event::Event (const Event &)`

25.104.2.3 `virtual gdcm::Event::~~Event () [virtual]`

25.104.3 Member Function Documentation

25.104.3.1 `virtual bool gdcm::Event::CheckEvent (const Event *) const [pure virtual]`

Check if given event matches or derives from this event.

25.104.3.2 `virtual const char* gdcm::Event::GetEventName (void) const [pure virtual]`

Return the StringName associated with the event.

Implemented in `gdcm::ProgressEvent`, `gdcm::DataSetEvent`, `gdcm::AnonymizeEvent`, and `gdcm::DataEvent`.

25.104.3.3 `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::ProgressEvent`, `gdcm::DataSetEvent`, `gdcm::AnonymizeEvent`, and `gdcm::DataEvent`.

25.104.3.4 `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:

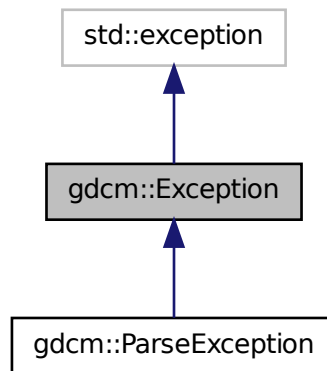
- gdcmEvent.h

25.105 gdcm::Exception Class Reference

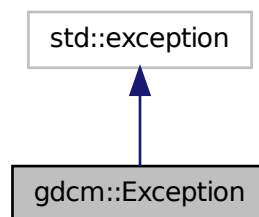
Exception.

```
#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

25.105.1 Detailed Description

Exception.

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

25.105.2 Constructor & Destructor Documentation

25.105.2.1 `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.

25.105.2.2 `virtual gdcm::Exception::~~Exception () throw () [inline],[virtual]`

25.105.3 Member Function Documentation

25.105.3.1 `const char* gdcm::Exception::GetDescription () const [inline]`

Return the Description.

Referenced by `gdcm::SequenceOfItems::Read()`.

25.105.3.2 `const char* gdcm::Exception::what () const throw () [inline]`

what implementation

Referenced by `gdcm::SequenceOfFragments::Read()`.

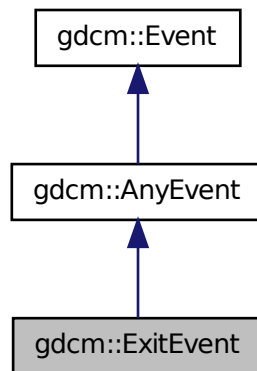
The documentation for this class was generated from the following file:

- `gdcmException.h`

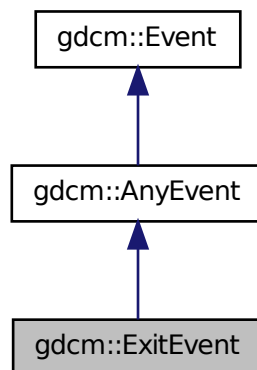
25.106 gdcm::ExitEvent Class Reference

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

Inheritance diagram for `gdcm::ExitEvent`:



Collaboration diagram for `gdcm::ExitEvent`:



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

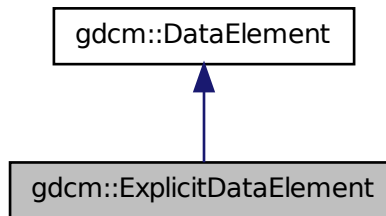
- `gdcmEvent.h`

25.107 `gdcm::ExplicitDataElement` Class Reference

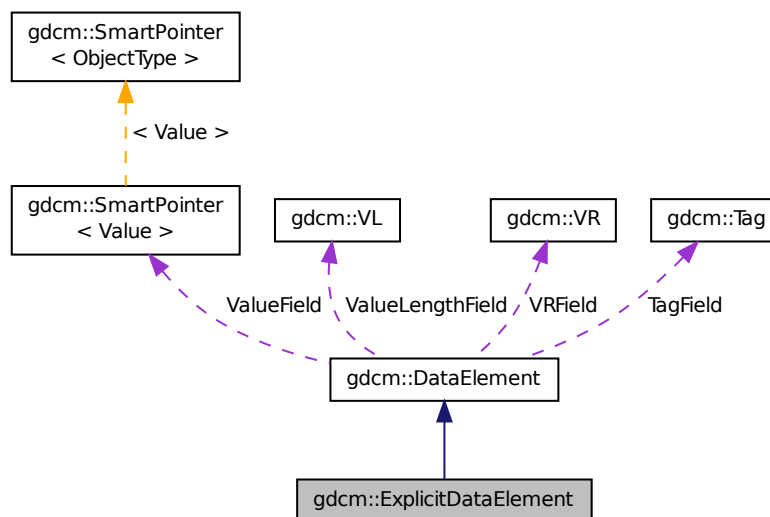
Class to read/write a `DataElement` as Explicit Data Element.

```
#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)

- `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

25.107.1 Detailed Description

Class to read/write a DataElement as Explicit Data Element.

Note

bla

25.107.2 Member Function Documentation

25.107.2.1 VL gdcm::ExplicitDataElement::GetLength () const

Reimplemented from `gdcm::DataElement`.

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

Reimplemented from `gdcm::DataElement`.

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

25.107.2.4 `template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadValue (std::istream & is)`

25.107.2.5 `template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

Reimplemented from `gdcm::DataElement`.

25.107.2.6 `template<typename TSwap > const std::ostream& gdcm::ExplicitDataElement::Write (std::ostream & os) const`

Reimplemented from `gdcm::DataElement`.

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

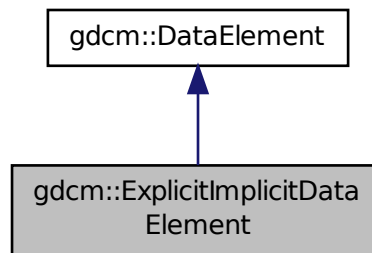
- `gdcmExplicitDataElement.h`

25.108 gdcm::ExplicitImplicitDataElement Class Reference

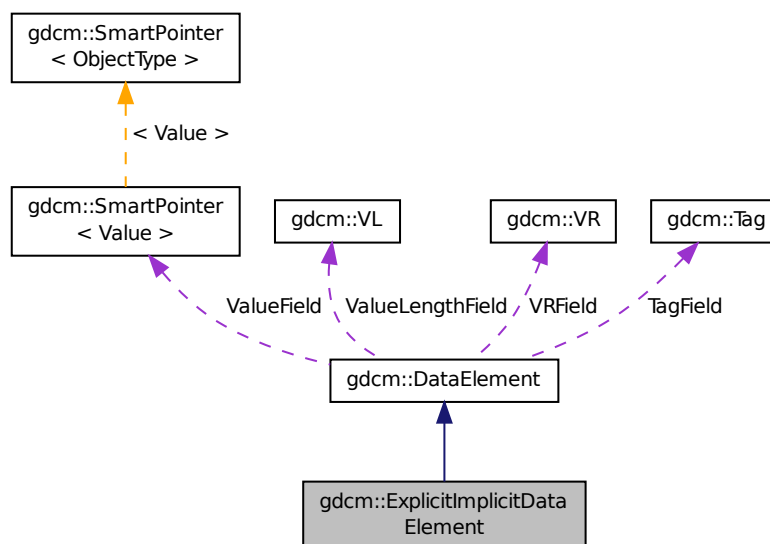
Class to read/write a DataElement as ExplicitImplicit Data Element.

```
#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)

- `template<typename TSwap >`
`std::istream & ReadWithLength (std::istream &is, VL &length)`

Additional Inherited Members

25.108.1 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.

25.108.2 Member Function Documentation

25.108.2.1 VL `gdcm::ExplicitImplicitDataElement::GetLength () const`

Reimplemented from `gdcm::DataElement`.

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

Reimplemented from `gdcm::DataElement`.

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

25.108.2.4 `template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadValue (std::istream & is)`

25.108.2.5 `template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadWithLength (std::istream & is, VL & length) [inline]`

Reimplemented from `gdcm::DataElement`.

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

- `gdcmExplicitImplicitDataElement.h`

25.109 gdcm::Fiducials Class Reference

Fiducials.

```
#include <gdcmFiducials.h>
```

Public Member Functions

- `Fiducials ()`

25.109.1 Detailed Description

Fiducials.

25.109.2 Constructor & Destructor Documentation

25.109.2.1 gdcM::Fiducials::Fiducials () [inline]

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

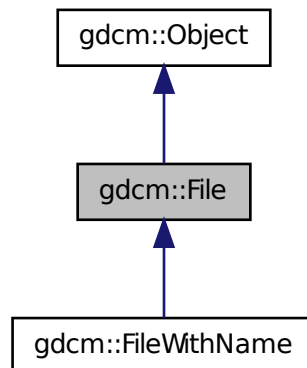
- gdcMfiducials.h

25.110 gdcM::File Class Reference

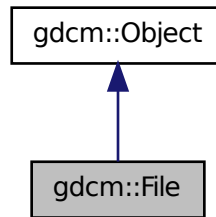
a DICOM File See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may be written, read and/or deleted.

```
#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

25.110.1 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:

ChangeSequenceUltrasound.cxx, CreateJPIPDataSet.cxx, DiffFile.cxx, DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, and StreamImageReaderTest.cxx.

25.110.2 Constructor & Destructor Documentation

25.110.2.1 `gdcm::File () [inline]`

25.110.2.2 `gdcm::File::~~File () [inline]`

25.110.3 Member Function Documentation

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

Get Data Set.

Examples:

ChangeSequenceUltrasound.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.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, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.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, and StreamImageReaderTest.cxx.

25.110.3.2 `DataSet& gdcm::File::GetDataSet () [inline]`

Get Data Set.

25.110.3.3 `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, MergeTwoFiles.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

Referenced by `gdcm::operator<<()`.

25.110.3.4 **FileMetaInformation& gdcm::File::GetHeader ()** [inline]

Get File Meta Information.

25.110.3.5 **std::istream& gdcm::File::Read (std::istream & is)**

Read.

25.110.3.6 **void gdcm::File::SetDataSet (const DataSet & ds)** [inline]

Set Data Set.

25.110.3.7 **void gdcm::File::SetHeader (const FileMetaInformation & fmi)** [inline]

Set File Meta Information.

25.110.3.8 **std::ostream const& gdcm::File::Write (std::ostream & os) const**

Write.

25.110.4 Friends And Related Function Documentation

25.110.4.1 **std::ostream& operator<< (std::ostream & os, const File & val)** [friend]

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

- gdcmFile.h

25.111 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 ()

25.111.1 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.

25.111.2 Constructor & Destructor Documentation

25.111.2.1 `gdcm::FileDerivation::FileDerivation ()`

25.111.2.2 `gdcm::FileDerivation::~~FileDerivation ()`

25.111.3 Member Function Documentation

25.111.3.1 `bool gdcm::FileDerivation::AddDerivationDescription ()` [protected]

25.111.3.2 `bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (DataSet & ds)` [protected]

25.111.3.3 `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.

25.111.3.4 `bool gdcM::FileDerivation::AddSourceImageSequence ()` [protected]

25.111.3.5 `bool gdcM::FileDerivation::Derive ()`

Change.

Examples:

GenFakelImage.cxx.

25.111.3.6 `File& gdcM::FileDerivation::GetFile ()` [inline]

Examples:

GenFakelImage.cxx.

25.111.3.7 `const File& gdcM::FileDerivation::GetFile () const` [inline]

25.111.3.8 `void gdcM::FileDerivation::SetDerivationCodeSequenceCodeValue (unsigned int codevalue)`

Specify the Derivation Code Sequence Code Value. Eg 113040.

Examples:

GenFakelImage.cxx.

25.111.3.9 `void gdcM::FileDerivation::SetDerivationDescription (const char * dd)`

Specify the Derivation Description. Eg "lossy conversion".

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

Set/Get File.

Examples:

GenFakelImage.cxx.

25.111.3.11 `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

25.112 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)`

25.112.1 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`.

25.112.2 Constructor & Destructor Documentation

25.112.2.1 `gdcm::FileExplicitFilter::FileExplicitFilter ()` `[inline]`

25.112.2.2 `gdcm::FileExplicitFilter::~~FileExplicitFilter ()` `[inline]`

25.112.3 Member Function Documentation

25.112.3.1 `bool gdcm::FileExplicitFilter::Change ()`

Set FMI Transfer Syntax.

Change

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

25.112.3.2 `bool gdcm::FileExplicitFilter::ChangeFMI ()` `[protected]`

25.112.3.3 `File& gdcm::FileExplicitFilter::GetFile ()` `[inline]`

25.112.3.4 `bool gdcm::FileExplicitFilter::ProcessDataSet (DataSet & ds, Dicts const & dicts)` `[protected]`

25.112.3.5 `void gdcm::FileExplicitFilter::SetChangePrivateTags (bool b)` `[inline]`

Decide whether or not to VR'ify private tags.

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

Set/Get File.

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

25.112.3.7 `void gdcm::FileExplicitFilter::SetRecomputeItemLength (bool b)`

By default set Sequence & Item length to Undefined to avoid recomputing length:

25.112.3.8 `void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (bool b)`

25.112.3.9 `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:

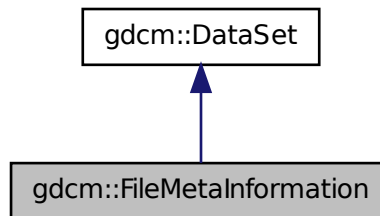
- `gdcmFileExplicitFilter.h`

25.113 gdcm::FileMetaInformation Class Reference

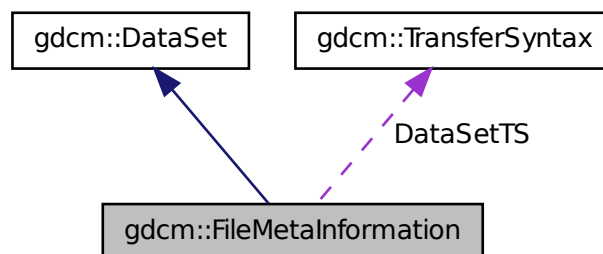
Class to represent a File Meta Information.

```
#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`
- `TransferSyntax::NegociatedType 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)`
Replace a dataelement with another one.
- `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

25.113.1 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, and ReadAndDumpDICOMDIR.cxx.

25.113.2 Constructor & Destructor Documentation

25.113.2.1 `gdcm::FileMetaInformation::FileMetaInformation ()` `[inline]`

25.113.2.2 `gdcm::FileMetaInformation::~~FileMetaInformation ()` `[inline]`

25.113.2.3 `gdcm::FileMetaInformation::FileMetaInformation (FileMetaInformation const & fmi)` `[inline]`

References DataSetMS, DataSetTS, and MetaInformationTS.

25.113.3 Member Function Documentation

25.113.3.1 `static void gdcm::FileMetaInformation::AppendImplementationClassUID (const char * imp)` `[static]`

25.113.3.2 `void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ()` `[protected]`

25.113.3.3 `void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ()` `[protected]`

25.113.3.4 `void gdcm::FileMetaInformation::Default ()` `[protected]`

25.113.3.5 `void gdcm::FileMetaInformation::FillFromDataSet (DataSet const & ds)`

Construct a FileMetaInformation from an already existing DataSet:

25.113.3.6 **const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax () const** [inline]

Examples:

GetJPEGSamplePrecision.cxx, and MergeTwoFiles.cxx.

25.113.3.7 **static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion ()** [static],[protected]

25.113.3.8 **VL gdcm::FileMetaInformation::GetFullLength () const** [inline]

References gdcm::VL::GetLength().

25.113.3.9 **static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID ()** [static],[protected]

25.113.3.10 **static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName ()** [static],[protected]

25.113.3.11 **static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ()** [static],[protected]

25.113.3.12 **static const char* gdcm::FileMetaInformation::GetImplementationClassUID ()** [static]

25.113.3.13 **static const char* gdcm::FileMetaInformation::GetImplementationVersionName ()** [static]

25.113.3.14 **MediaStorage gdcm::FileMetaInformation::GetMediaStorage () const**

25.113.3.15 **TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS () const** [inline]

25.113.3.16 **const Preamble& gdcm::FileMetaInformation::GetPreamble () const** [inline]

Get Preamble.

Referenced by gdcm::operator<<().

25.113.3.17 **Preamble& gdcm::FileMetaInformation::GetPreamble ()** [inline]

25.113.3.18 **static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ()** [static]

25.113.3.19 **void gdcm::FileMetaInformation::Insert (const DataElement & de)** [inline]

Insert a DataElement in the DataSet.

Warning

: Tag need to be >= 0x8 to be considered valid data element

Reimplemented from gdcm::DataSet.

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

25.113.3.20 `bool gdcm::FileMetaInformation::IsValid () const [inline]`

25.113.3.21 `std::istream& gdcm::FileMetaInformation::Read (std::istream & is)`

Read.

Reimplemented from `gdcm::DataSet`.

25.113.3.22 `std::istream& gdcm::FileMetaInformation::ReadCompat (std::istream & is)`

25.113.3.23 `template<typename TSwap > std::istream& gdcm::FileMetaInformation::ReadCompatInternal (std::istream & is)`
[protected]

25.113.3.24 `void gdcm::FileMetaInformation::Replace (const DataElement & de) [inline]`

Replace a dataelement with another one.

Reimplemented from `gdcm::DataSet`.

Examples:

LargeVRDSExplicit.cxx.

References `gdcm::DataElement::GetTag()`.

25.113.3.25 `void gdcm::FileMetaInformation::SetDataSetTransferSyntax (const TransferSyntax & ts)`

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixJAIBugJPEGs.cxx, GenAIIVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, pmsct_rgb1.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

25.113.3.26 `static void gdcm::FileMetaInformation::SetImplementationClassUID (const char * imp) [static]`

Override the GDCM default values:

25.113.3.27 `static void gdcm::FileMetaInformation::SetImplementationVersionName (const char * version) [static]`

25.113.3.28 `void gdcm::FileMetaInformation::SetPreamble (const Preamble & p) [inline]`

25.113.3.29 `static void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (const char * title) [static]`

Examples:

FixJAIBugJPEGs.cxx.

25.113.3.30 `std::ostream& gdcm::FileMetaInformation::Write (std::ostream & os) const`

Write.

Reimplemented from `gdcm::DataSet`.

25.113.4 Friends And Related Function Documentation

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

25.113.5 Member Data Documentation

25.113.5.1 `MediaStorage::MSType gdcmm::FileMetaInformation::DataSetMS` [protected]

Referenced by `FileMetaInformation()`.

25.113.5.2 `TransferSyntax gdcmm::FileMetaInformation::DataSetTS` [protected]

Referenced by `FileMetaInformation()`.

25.113.5.3 `TransferSyntax::NegociatedType gdcmm::FileMetaInformation::MetaInformationTS` [protected]

Referenced by `FileMetaInformation()`.

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

- `gdcmmFileMetaInformation.h`

25.114 gdcmm::Filename Class Reference

Class to manipulate file name's.

```
#include <gdcmmFilename.h>
```

Public Member Functions

- `Filename (const char *filename="")`
- `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 foward slash (UNIX style) to windows style slash.

Static Public Member Functions

- static const char * Join (const char *path, const char *filename)

25.114.1 Detailed Description

Class to manipulate file name's.

Note

OS independant representation of a filename (to query path, name and extension from a filename)

25.114.2 Constructor & Destructor Documentation

25.114.2.1 `gdcm::Filename::Filename (const char * filename = " ") [inline]`

25.114.3 Member Function Documentation

25.114.3.1 `const char* gdcm::Filename::GetExtension ()`

return only the extension part of a filename

25.114.3.2 `const char* gdcm::Filename::GetFileName () const [inline]`

Return the full filename.

25.114.3.3 `const char* gdcm::Filename::GetName ()`

return only the name part of a filename

25.114.3.4 `const char* gdcm::Filename::GetPath ()`

Return only the path component of a filename.

25.114.3.5 `bool gdcm::Filename::IsEmpty () const [inline]`

return whether the filename is empty

25.114.3.6 `bool gdcm::Filename::IsIdentical (Filename const & fn) const`

25.114.3.7 `static const char* gdcm::Filename::Join (const char * path, const char * filename) [static]`

Join two paths NOT THREAD SAFE

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

Simple operator to allow Filename myfilename("..."); const char * s = myfilename;

25.114.3.9 `const char* gdcm::Filename::ToUnixSlashes ()`

Convert backslash (windows style) to UNIX style slash.

25.114.3.10 `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`

25.115 `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.

25.115.1 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`.

25.115.2 Member Typedef Documentation

25.115.2.1 `typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType`

25.115.2.2 `typedef std::string gdcm::FilenameGenerator::FilenameType`

25.115.2.3 `typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType`

25.115.3 Constructor & Destructor Documentation

25.115.3.1 `gdcm::FilenameGenerator::FilenameGenerator ()` `[inline]`

25.115.3.2 `gdcm::FilenameGenerator::~~FilenameGenerator ()` `[inline]`

25.115.4 Member Function Documentation

25.115.4.1 `bool gdcm::FilenameGenerator::Generate ()`

Generate (return success)

Examples:

`ConvertMultiFrameToSingleFrame.cxx`.

25.115.4.2 `const char* gdcm::FilenameGenerator::GetFilename (SizeType n) const`

Get a particular filename (call after `Generate`)

Examples:

`ConvertMultiFrameToSingleFrame.cxx`.

25.115.4.3 `FilenamesType const& gdcm::FilenameGenerator::GetFilenames () const` `[inline]`

Return all filenames.

25.115.4.4 **SizeType** gdcm::FilenameGenerator::GetNumberOfFileNames () const

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.4.5 const char* gdcm::FilenameGenerator::GetPattern () const [inline]

25.115.4.6 const char* gdcm::FilenameGenerator::GetPrefix () const [inline]

25.115.4.7 void gdcm::FilenameGenerator::SetNumberOfFileNames (**SizeType** *nfiles*)

Set/Get the number of filenames to generate.

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.4.8 void gdcm::FilenameGenerator::SetPattern (const char * *pattern*) [inline]

Set/Get pattern.

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.115.4.9 void gdcm::FilenameGenerator::SetPrefix (const char * *prefix*) [inline]

Set/Get prefix.

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

- gdcmFilenameGenerator.h

25.116 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)
- FileType const & GetFiles () const
- void SetFiles (FileType const &files)

Friends

- std::ostream & operator<< (std::ostream &_os, const FileSet &d)

25.116.1 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.

25.116.2 Member Typedef Documentation

25.116.2.1 `typedef std::vector<FileType> gdcm::FileSet::FileType`

25.116.2.2 `typedef std::string gdcm::FileSet::FileType`

25.116.3 Constructor & Destructor Documentation

25.116.3.1 `gdcm::FileSet::FileSet ()` `[inline]`

25.116.4 Member Function Documentation

25.116.4.1 `void gdcm::FileSet::AddFile (File const &)` `[inline]`

Deprecated . Does nothing

25.116.4.2 `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.

25.116.4.3 `FileType const& gdcm::FileSet::GetFiles () const` `[inline]`

25.116.4.4 `void gdcm::FileSet::SetFiles (FileType const & files)`

25.116.5 Friends And Related Function Documentation

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

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

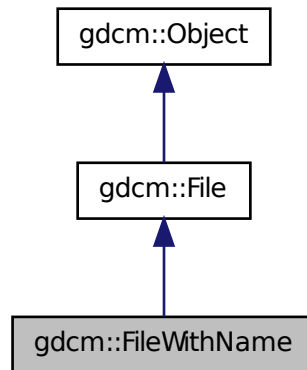
- gdcmFileSet.h

25.117 gdcm::FileWithName Class Reference

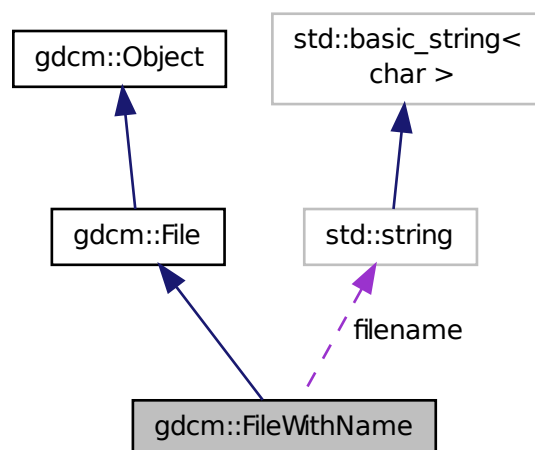
FileWithName.

```
#include <gdcmSerieHelper.h>
```

Inheritance diagram for gdcm::FileWithName:



Collaboration diagram for gdcm::FileWithName:



Public Member Functions

- `FileName` (File &f)

Public Attributes

- `std::string filename`

25.117.1 Detailed Description

`FileName`.

Backward only class do not use in newer code

25.117.2 Constructor & Destructor Documentation

25.117.2.1 `gdcm::FileName::FileName (File & f)` `[inline]`

25.117.3 Member Data Documentation

25.117.3.1 `std::string gdcm::FileName::filename`

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

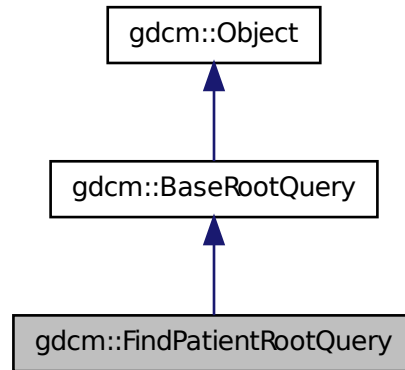
- `gdcmSerieHelper.h`

25.118 gdcm::FindPatientRootQuery Class Reference

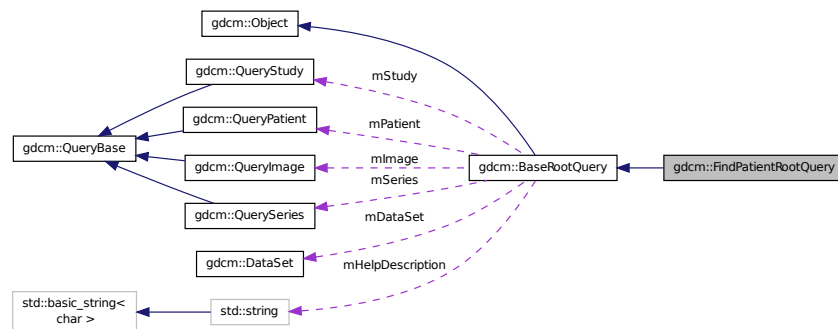
`FindPatientRootQuery` contains: the class which will produce a dataset for c-find with patient root.

```
#include <gdcmFindPatientRootQuery.h>
```

Inheritance diagram for `gdcmm::FindPatientRootQuery`:



Collaboration diagram for `gdcmm::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

25.118.1 Detailed Description

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

25.118.2 Constructor & Destructor Documentation

25.118.2.1 `gdcm::FindPatientRootQuery::FindPatientRootQuery ()`

25.118.3 Member Function Documentation

25.118.3.1 `UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID () const` `[virtual]`

Implements `gdcm::BaseRootQuery`.

25.118.3.2 `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`.

25.118.3.3 `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`.

25.118.3.4 `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`.

25.118.4 Friends And Related Function Documentation

25.118.4.1 `friend class QueryFactory` `[friend]`

Reimplemented from `gdcm::BaseRootQuery`.

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

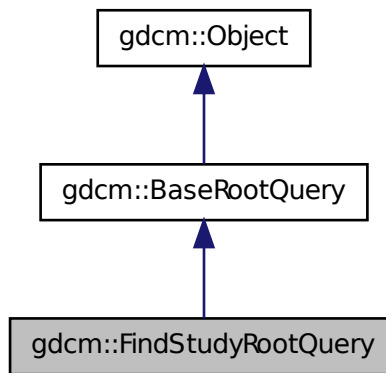
- `gdcmFindPatientRootQuery.h`

25.119 `gdcm::FindStudyRootQuery` Class Reference

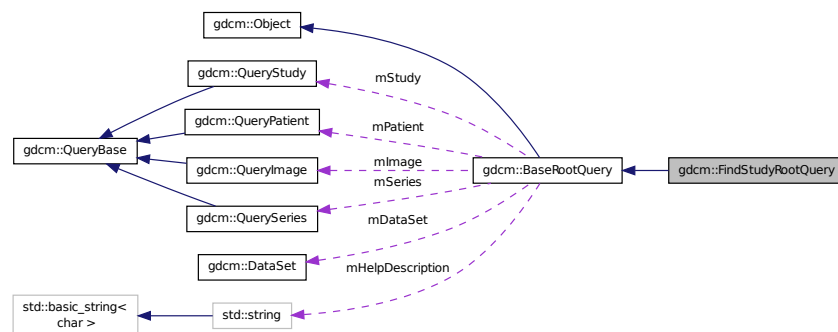
`FindStudyRootQuery` contains: the class which will produce a dataset for C-FIND with study root.

```
#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

25.119.1 Detailed Description

FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

25.119.2 Constructor & Destructor Documentation

25.119.2.1 `gdcm::FindStudyRootQuery::FindStudyRootQuery ()`

25.119.3 Member Function Documentation

25.119.3.1 `UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID () const` [virtual]

Implements `gdcm::BaseRootQuery`.

25.119.3.2 `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`.

25.119.3.3 `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`.

25.119.3.4 `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`.

25.119.4 Friends And Related Function Documentation

25.119.4.1 friend class **QueryFactory** [friend]

Reimplemented from `gdcm::BaseRootQuery`.

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

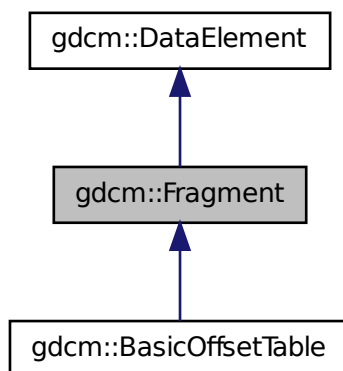
- `gdcmFindStudyRootQuery.h`

25.120 `gdcm::Fragment` Class Reference

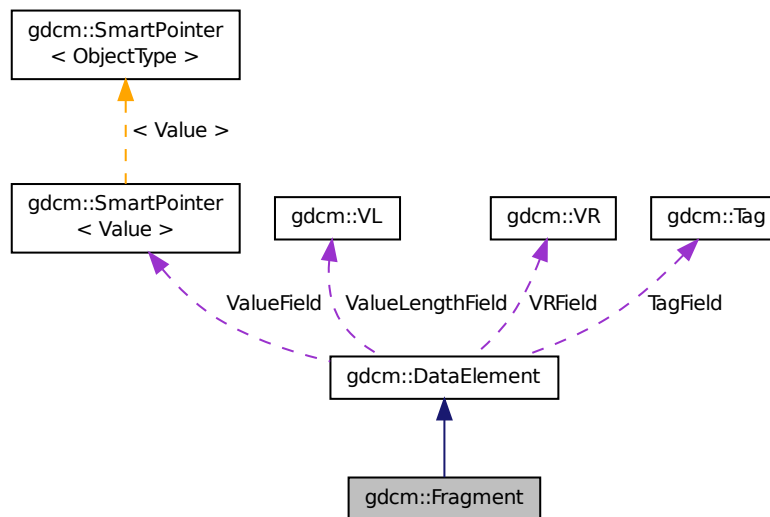
Class to represent a Fragment.

```
#include <gdcmFragment.h>
```

Inheritance diagram for `gdcm::Fragment`:



Collaboration diagram for gdcm::Fragment:



Public Member Functions

- `Fragment ()`
- `VL GetLength () const`
- `template<typename TSwap > std::istream & Read (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

25.120.1 Detailed Description

Class to represent a Fragment.

Examples:

`FixBrokenJ2K.cxx`, and `FixJAIBugJPEGLS.cxx`.

25.120.2 Constructor & Destructor Documentation

25.120.2.1 `gdcm::Fragment::Fragment ()` `[inline]`

25.120.3 Member Function Documentation

25.120.3.1 `VL gdcm::Fragment::GetLength () const` `[inline]`

Reimplemented from `gdcm::DataElement`.

References `gdcm::VL::GetLength()`.

25.120.3.2 `template<typename TSwap > std::istream& gdcm::Fragment::Read (std::istream & is)` `[inline]`

Reimplemented from `gdcm::DataElement`.

Reimplemented in `gdcm::BasicOffsetTable`.

References `gdcm::VL::Read()`.

Referenced by `gdcm::SequenceOfFragments::Read()`.

25.120.3.3 `template<typename TSwap > std::istream& gdcm::Fragment::ReadValue (std::istream & is)` `[inline]`

References `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

25.120.3.4 `template<typename TSwap > std::ostream& gdcm::Fragment::Write (std::ostream & os) const` `[inline]`

Reimplemented from `gdcm::DataElement`.

References `gdcm::ByteValue::GetLength()`, `gdcm::VL::Write()`, and `gdcm::ByteValue::Write()`.

25.120.4 Friends And Related Function Documentation

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

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

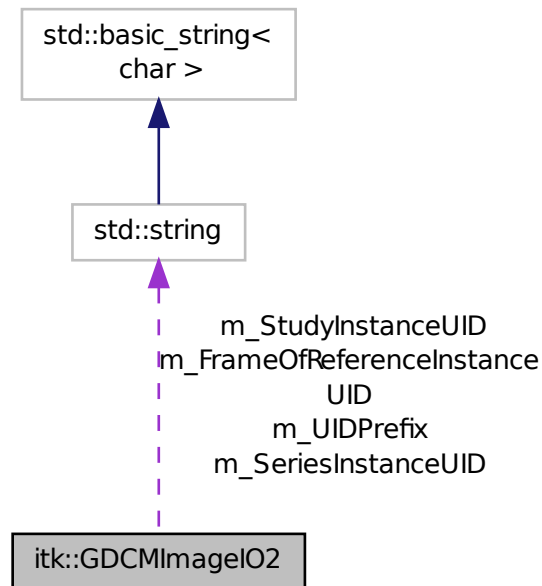
- `gdcmFragment.h`

25.121 itk::GDCMImageIO2 Class Reference

ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the `gdcm` library (currently `gdcm 2.0` is used):

```
#include <itkGDCMImageIO2.h>
```

Collaboration diagram for itk::GDCMImageIO2:



Public Types

- typedef SmartPointer< Self > Pointer
- typedef GDCMImageIO2 Self
- typedef ImageIOBase Superclass
- enum TCompressionType {
 JPEG = 0,
 JPEG2000 }

Public Member Functions

- virtual bool CanReadFile (const char *)
- virtual bool CanWriteFile (const char *)
- void GetBodyPart (char *part)
- void GetInstitution (char *ins)
- void GetManufacturer (char *manu)
- void GetModality (char *modality)
- void GetModel (char *model)
- void GetNumberOfSeriesInStudy (char *series)
- void GetNumberOfStudyRelatedSeries (char *series)
- void GetPatientAge (char *age)
- void GetPatientDOB (char *dob)
- void GetPatientID (char *id)

- void GetPatientName (char *name)
- void GetPatientSex (char *sex)
- void GetScanOptions (char *options)
- void GetStudyDate (char *date)
- void GetStudyDescription (char *desc)
- void GetStudyID (char *id)
- bool GetValueFromTag (const std::string &tag, std::string &value)
- itkBooleanMacro (KeepOriginalUID)
- itkBooleanMacro (LoadSequences)
- itkBooleanMacro (LoadPrivateTags)
- itkGetEnumMacro (CompressionType, TCompressionType)
- itkGetMacro (RescaleSlope, double)
- itkGetMacro (RescaleIntercept, double)
- itkGetMacro (KeepOriginalUID, bool)
- itkGetMacro (LoadSequences, bool)
- itkGetMacro (LoadPrivateTags, bool)
- itkGetStringMacro (UIDPrefix)
- itkGetStringMacro (StudyInstanceUID)
- itkGetStringMacro (SeriesInstanceUID)
- itkGetStringMacro (FrameOfReferenceInstanceUID)
- itkNewMacro (Self)
- itkSetEnumMacro (CompressionType, TCompressionType)
- itkSetMacro (KeepOriginalUID, bool)
- itkSetMacro (MaxSizeLoadEntry, long)
- itkSetMacro (LoadSequences, bool)
- itkSetMacro (LoadPrivateTags, bool)
- itkSetStringMacro (UIDPrefix)
- itkTypeMacro (GDCMImageIO2, Superclass)
- virtual void Read (void *buffer)
- virtual void ReadImageInformation ()
- virtual void Write (const void *buffer)
- virtual void WriteImageInformation ()

Static Public Member Functions

- static bool GetLabelFromTag (const std::string &tag, std::string &labelId)
- static bool GetLoadPrivateTagsDefault ()
- static bool GetLoadSequencesDefault ()
- static void LoadPrivateTagsDefaultOff ()
- static void LoadPrivateTagsDefaultOn ()
- static void LoadSequencesDefaultOff ()
- static void LoadSequencesDefaultOn ()
- static void SetLoadPrivateTagsDefault (bool)
- static void SetLoadSequencesDefault (bool)

Protected Member Functions

- GDCMImageIO2 ()
- ~GDCMImageIO2 ()
- void InternalReadImageInformation (std::ifstream &)
- bool OpenGDCMFileForReading (std::ifstream &, const char *)
- bool OpenGDCMFileForWriting (std::ofstream &, const char *)
- void PrintSelf (std::ostream &os, Indent indent) const

Protected Attributes

- std::string m_FrameOfReferenceInstanceUID
- bool m_KeepOriginalUID
- double m_RescaleIntercept
- double m_RescaleSlope
- std::string m_SeriesInstanceUID
- std::string m_StudyInstanceUID
- std::string m_UIDPrefix

25.121.1 Detailed Description

ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the gdcm library (currently gdcm 2.0 is used):

<http://gdcm.sourceforge.net>

\warning this class is deprecated, as gdcm 2.x has been integrated in ITK starting ITK 3.12

25.121.2 Member Typedef Documentation

25.121.2.1 `typedef SmartPointer<Self> itk::GDCMImageIO2::Pointer`

25.121.2.2 `typedef GDCMImageIO2 itk::GDCMImageIO2::Self`

Standard class typedefs.

25.121.2.3 `typedef ImageIOBase itk::GDCMImageIO2::Superclass`

25.121.3 Member Enumeration Documentation

25.121.3.1 `enum itk::GDCMImageIO2::TCompressionType`

Set/Get a boolean to use the JPEG2000 compression or not.

Enumerator:

JPEG

JPEG2000

25.121.4 Constructor & Destructor Documentation

25.121.4.1 itk::GDCMImageIO2::GDCMImageIO2 () [protected]

25.121.4.2 itk::GDCMImageIO2::~~GDCMImageIO2 () [protected]

25.121.5 Member Function Documentation

25.121.5.1 virtual bool itk::GDCMImageIO2::CanReadFile (const char *) [virtual]

Determine the file type. Returns true if this ImageIO can read the file specified.

25.121.5.2 virtual bool itk::GDCMImageIO2::CanWriteFile (const char *) [virtual]

Determine the file type. Returns true if this ImageIO can write the file specified. GDCM triggers on ".dcm" and ".dicom".

25.121.5.3 void itk::GDCMImageIO2::GetBodyPart (char * *part*)

25.121.5.4 void itk::GDCMImageIO2::GetInstitution (char * *ins*)

25.121.5.5 static bool itk::GDCMImageIO2::GetLabelFromTag (const std::string & *tag*, std::string & *labelId*) [static]

Method for consulting the DICOM dictionary and recovering the text description of a field using its numeric tag represented as a string. If the tagkey is not found in the dictionary then this static method return false and the value "Unknown" in the labelId. If the tagkey is found then this static method returns true and the actual string descriptor of the tagkey is returned in the variable labelId.

25.121.5.6 static bool itk::GDCMImageIO2::GetLoadPrivateTagsDefault () [inline],[static]

25.121.5.7 static bool itk::GDCMImageIO2::GetLoadSequencesDefault () [inline],[static]

25.121.5.8 void itk::GDCMImageIO2::GetManufacturer (char * *manu*)

25.121.5.9 void itk::GDCMImageIO2::GetModality (char * *modality*)

25.121.5.10 void itk::GDCMImageIO2::GetModel (char * *model*)

25.121.5.11 void itk::GDCMImageIO2::GetNumberOfSeriesInStudy (char * *series*)

25.121.5.12 void itk::GDCMImageIO2::GetNumberOfStudyRelatedSeries (char * *series*)

25.121.5.13 void itk::GDCMImageIO2::GetPatientAge (char * *age*)

25.121.5.14 void itk::GDCMImageIO2::GetPatientDOB (char * *dob*)

25.121.5.15 void itk::GDCMImageIO2::GetPatientID (char * *id*)

25.121.5.16 void itk::GDCMImageIO2::GetPatientName (char * *name*)

Convenience methods to query patient information and scanner information. These methods are here for compatibility with the DICOMImageIO2 class.

25.121.5.17 void itk::GDCMImageIO2::GetPatientSex (char * *sex*)

25.121.5.18 void itk::GDCMImageIO2::GetScanOptions (char * *options*)

25.121.5.19 void itk::GDCMImageIO2::GetStudyDate (char * *date*)

25.121.5.20 void itk::GDCMImageIO2::GetStudyDescription (char * *desc*)

25.121.5.21 void itk::GDCMImageIO2::GetStudyID (char * *id*)

25.121.5.22 bool itk::GDCMImageIO2::GetValueFromTag (const std::string & *tag*, std::string & *value*)

More general method to retrieve an arbitrary DICOM value based on a DICOM Tag (eg "0123|4567"). WARNING: You need to use the lower case for hex 0x[a-f], for instance: "0020|000d" instead of "0020|000D" (the latter won't work)

25.121.5.23 void itk::GDCMImageIO2::InternalReadImageInformation (std::ifstream &) [protected]

25.121.5.24 itk::GDCMImageIO2::itkBooleanMacro (KeepOriginalUID)

25.121.5.25 itk::GDCMImageIO2::itkBooleanMacro (LoadSequences)

25.121.5.26 itk::GDCMImageIO2::itkBooleanMacro (LoadPrivateTags)

25.121.5.27 itk::GDCMImageIO2::itkGetEnumMacro (CompressionType , TCompressionType)

25.121.5.28 itk::GDCMImageIO2::itkGetMacro (RescaleSlope , double)

Macro to access Rescale Slope and Rescale Intercept. Which are needed to rescale properly image when needed. User then need to Always check those value when access value from the DICOM header

25.121.5.29 itk::GDCMImageIO2::itkGetMacro (RescaleIntercept , double)

25.121.5.30 itk::GDCMImageIO2::itkGetMacro (KeepOriginalUID , bool)

25.121.5.31 itk::GDCMImageIO2::itkGetMacro (LoadSequences , bool)

25.121.5.32 itk::GDCMImageIO2::itkGetMacro (LoadPrivateTags , bool)

25.121.5.33 itk::GDCMImageIO2::itkGetStringMacro (UIDPrefix)

Macro to access the DICOM UID prefix. By default this is the ITK root id. This default can be overridden if the exam is for example part of an existing study.

25.121.5.34 `itk::GDCMImageIO2::itkGetStringMacro (StudyInstanceUID)`

Access the generated DICOM UID's.

25.121.5.35 `itk::GDCMImageIO2::itkGetStringMacro (SeriesInstanceUID)`

25.121.5.36 `itk::GDCMImageIO2::itkGetStringMacro (FrameOfReferenceInstanceUID)`

25.121.5.37 `itk::GDCMImageIO2::itkNewMacro (Self)`

Method for creation through the object factory.

25.121.5.38 `itk::GDCMImageIO2::itkSetEnumMacro (CompressionType , TCompressionType)`

25.121.5.39 `itk::GDCMImageIO2::itkSetMacro (KeepOriginalUID , bool)`

Preserve the original DICOM UID of the input files

25.121.5.40 `itk::GDCMImageIO2::itkSetMacro (MaxSizeLoadEntry , long)`

A DICOM file can contains multiple binary stream that can be very long For example an Overlay on the image. Most of the time user do not want to load this binary structure in memory since it can consume lot of memory. Therefore any field that is bigger than the default value 0xffff is discarded and just seek'd This method allow advanced user to force the reading of such field

25.121.5.41 `itk::GDCMImageIO2::itkSetMacro (LoadSequences , bool)`

Parse any sequences in the DICOM file. Defaults to the value of LoadSequencesDefault. Loading DICOM files is faster when sequences are not needed.

25.121.5.42 `itk::GDCMImageIO2::itkSetMacro (LoadPrivateTags , bool)`

Parse any private tags in the DICOM file. Defaults to the value of LoadPrivateTagsDefault. Loading DICOM files is faster when private tags are not needed.

25.121.5.43 `itk::GDCMImageIO2::itkSetStringMacro (UIDPrefix)`

25.121.5.44 `itk::GDCMImageIO2::itkTypeMacro (GDCMImageIO2 , Superclass)`

Run-time type information (and related methods).

25.121.5.45 `static void itk::GDCMImageIO2::LoadPrivateTagsDefaultOff () [inline],[static]`

25.121.5.46 `static void itk::GDCMImageIO2::LoadPrivateTagsDefaultOn () [inline],[static]`

25.121.5.47 `static void itk::GDCMImageIO2::LoadSequencesDefaultOff () [inline],[static]`

25.121.5.48 `static void itk::GDCMImageIO2::LoadSequencesDefaultOn () [inline],[static]`

25.121.5.49 `bool itk::GDCMImageIO2::OpenGDCMFileForReading (std::ifstream & , const char *) [protected]`

25.121.5.50 `bool itk::GDCMImageIO2::OpenGDCMFileForWriting (std::ofstream & , const char *) [protected]`

25.121.5.51 `void itk::GDCMImageIO2::PrintSelf (std::ostream & os, Indent indent) const [protected]`

25.121.5.52 `virtual void itk::GDCMImageIO2::Read (void * buffer) [virtual]`

Reads the data from disk into the memory buffer provided.

25.121.5.53 `virtual void itk::GDCMImageIO2::ReadImageInformation () [virtual]`

Set the spacing and dimesion information for the current filename.

25.121.5.54 `static void itk::GDCMImageIO2::SetLoadPrivateTagsDefault (bool) [inline],[static]`

Global method to define the default value for LoadPrivateTags. When instances of GDCMImageIO are created, the ivar LoadPrivateTags is initialized to the value of LoadPrivateTagsDefault. This method is useful when relying on the IO factory mechanism to load images rather than specifying a particular ImageIO object on the readers. Default is false.

25.121.5.55 `static void itk::GDCMImageIO2::SetLoadSequencesDefault (bool) [inline],[static]`

Global method to define the default value for LoadSequences. When instances of GDCMImageIO are created, the ivar LoadSequences is initialized to the value of LoadSequencesDefault. This method is useful when relying on the IO factory mechanism to load images rather than specifying a particular ImageIO object on the readers. Default is false.

25.121.5.56 `virtual void itk::GDCMImageIO2::Write (const void * buffer) [virtual]`

Writes the data to disk from the memory buffer provided. Make sure that the IORegion has been set properly.

25.121.5.57 `virtual void itk::GDCMImageIO2::WriteImageInformation () [virtual]`

Writes the spacing and dimentions of the image. Assumes SetFileName has been called with a valid file name.

25.121.6 Member Data Documentation

25.121.6.1 `std::string itk::GDCMImageIO2::m_FrameOfReferenceInstanceUID [protected]`

25.121.6.2 `bool itk::GDCMImageIO2::m_KeepOriginalUID [protected]`

25.121.6.3 `double itk::GDCMImageIO2::m_RescaleIntercept [protected]`

25.121.6.4 `double itk::GDCMImageIO2::m_RescaleSlope [protected]`

25.121.6.5 `std::string itk::GDCMImageIO2::m_SeriesInstanceUID [protected]`

25.121.6.6 `std::string itk::GDCMImageIO2::m_StudyInstanceUID` [protected]

25.121.6.7 `std::string itk::GDCMImageIO2::m_UIDPrefix` [protected]

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

- `itkGDCMImageIO2.h`

25.122 `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 ressource file.

Friends

- `std::ostream & operator<< (std::ostream &_os, const Global &g)`

25.122.1 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, GenFakeIdentifyFile.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

25.122.2 Constructor & Destructor Documentation**25.122.2.1 gdcmm::Global::Global ()****25.122.2.2 gdcmm::Global::~~Global ()****25.122.3 Member Function Documentation****25.122.3.1 bool gdcmm::Global::Append (const char * *path*)**

Append path at the end of the path list

Warning

not thread safe !

25.122.3.2 Defs const& gdcmm::Global::GetDefs () const

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

Examples:

GenerateStandardSOPClasses.cxx.

25.122.3.3 Dicts const& gdcmm::Global::GetDicts () const

retrieve the default/internal dicts (Part 6) This dict is filled up at load time

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, MrProtocol.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

25.122.3.4 Dicts& gdcmm::Global::GetDicts ()**25.122.3.5 static Global& gdcmm::Global::GetInstance () [static]**

return the singleton instance

Examples:

GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, MrProtocol.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

25.122.3.6 `bool gdcmm::Global::LoadResourcesFiles ()`

Load all internal XML files, ressource path need to have been set before calling this member function (see Append/-Prepend members func)

Warning

not thread safe !

Examples:

GenerateStandardSOPClasses.cxx.

25.122.3.7 `const char* gdcmm::Global::Locate (const char * resfile) const` [protected]

Locate a ressource file.

25.122.3.8 `bool gdcmm::Global::Prepend (const char * path)`

Prepend path at the begining of the path list

Warning

not thread safe !

25.122.4 Friends And Related Function Documentation

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

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

- gdcmmGlobal.h

25.123 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)

25.123.1 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 ?

25.123.2 Member Typedef Documentation

25.123.2.1 `typedef std::vector<std::string> gdcmm::GroupDict::GroupStringVector`

25.123.3 Constructor & Destructor Documentation

25.123.3.1 `gdcmm::GroupDict::GroupDict ()` `[inline]`

25.123.3.2 `gdcmm::GroupDict::~~GroupDict ()` `[inline]`

25.123.4 Member Function Documentation

25.123.4.1 `void gdcmm::GroupDict::Add (std::string const & abbreviation, std::string const & name)` `[protected]`

25.123.4.2 `std::string const& gdcmm::GroupDict::GetAbbreviation (uint16_t num) const`

Referenced by `gdcmm::operator<<()`.

25.123.4.3 `std::string const& gdcmm::GroupDict::GetName (uint16_t num) const`

Referenced by `gdcmm::operator<<()`.

25.123.4.4 `void gdcmm::GroupDict::Insert (uint16_t num, std::string const & abbreviation, std::string const & name)`
`[protected]`

25.123.4.5 `size_t gdcmm::GroupDict::Size () const` `[inline]`

Referenced by `gdcmm::operator<<()`.

25.123.5 Friends And Related Function Documentation

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

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

- `gdcmGroupDict.h`

25.124 gdcm::IconImageFilter Class Reference

IconImageFilter This filter will extract icons from a `gdcm::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 ()`

25.124.1 Detailed Description

IconImageFilter This filter will extract icons from a `gdcm::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.

25.124.2 Constructor & Destructor Documentation

25.124.2.1 `gdcm::IconImageFilter::IconImageFilter ()`

25.124.2.2 `gdcm::IconImageFilter::~~IconImageFilter ()`

25.124.3 Member Function Documentation

25.124.3.1 `bool gdcm::IconImageFilter::Extract ()`

Extract all Icon found in File.

Examples:

ExtractIconFromFile.cxx.

25.124.3.2 `void gdcm::IconImageFilter::ExtractIconImages ()` [protected]

25.124.3.3 `void gdcm::IconImageFilter::ExtractVeprolIconImages ()` [protected]

25.124.3.4 `File& gdcm::IconImageFilter::GetFile ()` [inline]

25.124.3.5 `const File& gdcm::IconImageFilter::GetFile () const` [inline]

25.124.3.6 `IconImage& gdcm::IconImageFilter::GetIconImage (unsigned int i) const`

Examples:

ExtractIconFromFile.cxx.

25.124.3.7 unsigned int gdcmlIconImageFilter::GetNumberOfIconImages () const

Retrieve extract IconImage (need to call Extract first)

Examples:

ExtractIconFromFile.cxx.

25.124.3.8 void gdcmlIconImageFilter::SetFile (const File & f) [inline]

Set/Get File.

Examples:

ExtractIconFromFile.cxx.

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

- gdcmlIconImageFilter.h

25.125 gdcmlIconImageGenerator Class Reference

IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of gdcml::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

```
#include <gdcmlIconImageGenerator.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.*

25.125.1 Detailed Description

IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of `gdcm::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`.

25.125.2 Constructor & Destructor Documentation

25.125.2.1 `gdcm::IconImageGenerator::IconImageGenerator ()`

25.125.2.2 `gdcm::IconImageGenerator::~~IconImageGenerator ()`

25.125.3 Member Function Documentation

25.125.3.1 `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`.

25.125.3.2 `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. true, false generates invalid Icon Image Sequence

25.125.3.3 `bool gdcm::IconImageGenerator::Generate ()`

Generate Icon.

Examples:

`ExtractIconFromFile.cxx`.

25.125.3.4 `const IconImage& gdcm::IconImageGenerator::GetIconImage () const` `[inline]`

Retrieve generated Icon.

Examples:

ExtractIconFromFile.cxx.

25.125.3.5 `Pixmap& gdcm::IconImageGenerator::GetPixmap ()` `[inline]`

25.125.3.6 `const Pixmap& gdcm::IconImageGenerator::GetPixmap () const` `[inline]`

25.125.3.7 `void gdcm::IconImageGenerator::SetOutputDimensions (const unsigned int dims[2])`

Set Target dimension of output Icon.

Examples:

ExtractIconFromFile.cxx.

25.125.3.8 `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)`

25.125.3.9 `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.

25.125.3.10 `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`

25.126 `gdcm::ignore_char` Struct Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- `ignore_char (char c)`

Public Attributes

- char m_char

25.126.1 Constructor & Destructor Documentation

25.126.1.1 gdcm::ignore_char::ignore_char (char c) [inline]

25.126.2 Member Data Documentation

25.126.2.1 char gdcm::ignore_char::m_char

Referenced by gdcm::operator>>().

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

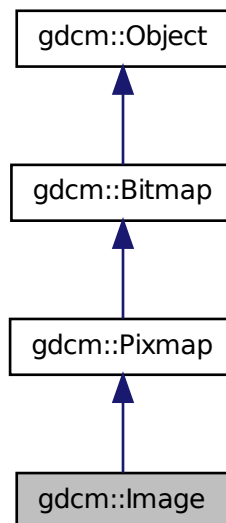
- gdcmElement.h

25.127 gdcm::Image Class Reference

Image.

```
#include <gdcmImage.h>
```

Inheritance diagram for gdcm::Image:



Note

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 seem redundant. One way to solve that would be to subclass gdcm::Image with gdcm::JPEGImage which would from the stream extract the header info and fill it to please gdcm::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, GenFakelImage.cxx, GetJPEGSamplePrecision.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

25.127.2 Constructor & Destructor Documentation

25.127.2.1 `gdcm::Image::Image ()` `[inline]`

25.127.2.2 `gdcm::Image::~~Image ()` `[inline]`

25.127.3 Member Function Documentation

25.127.3.1 `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.

25.127.3.2 `double gdcm::Image::GetDirectionCosines (unsigned int idx)` `const`

25.127.3.3 `double gdcm::Image::GetIntercept ()` `const` `[inline]`

25.127.3.4 `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.

25.127.3.5 `double gdcM::Image::GetOrigin (unsigned int idx) const`

25.127.3.6 `double gdcM::Image::GetSlope () const` `[inline]`

25.127.3.7 `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

25.127.3.8 `double gdcM::Image::GetSpacing (unsigned int idx) const`

25.127.3.9 `void gdcM::Image::Print (std::ostream & os) const` `[virtual]`

print

Reimplemented from `gdcM::Pixmap`.

Examples:

`CompressImage.cxx`, and `PatchFile.cxx`.

25.127.3.10 `void gdcM::Image::SetDirectionCosines (const float * dircos)`

25.127.3.11 `void gdcM::Image::SetDirectionCosines (const double * dircos)`

25.127.3.12 `void gdcM::Image::SetDirectionCosines (unsigned int idx, double dircos)`

25.127.3.13 `void gdcM::Image::SetIntercept (double intercept)` `[inline]`

intercept

25.127.3.14 `void gdcM::Image::SetOrigin (const float * ori)`

25.127.3.15 `void gdcM::Image::SetOrigin (const double * ori)`

25.127.3.16 `void gdcM::Image::SetOrigin (unsigned int idx, double ori)`

25.127.3.17 `void gdcM::Image::SetSlope (double slope)` `[inline]`

slope

25.127.3.18 `void gdcM::Image::SetSpacing (const double * spacing)`

Examples:

`csa2img.cxx`, and `iU22tomultisc.cxx`.

25.127.3.19 void gdcm::Image::SetSpacing (unsigned int *idx*, double *spacing*)

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

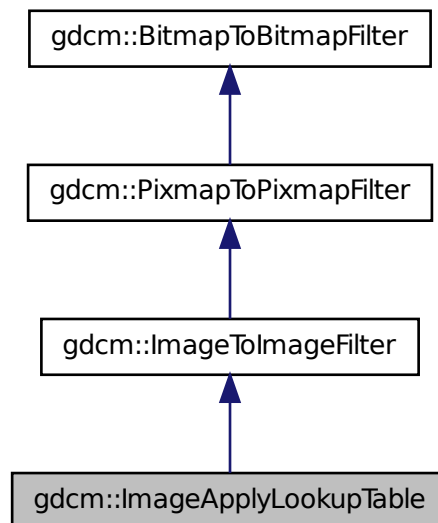
- gdcmImage.h

25.128 gdcm::ImageApplyLookupTable Class Reference

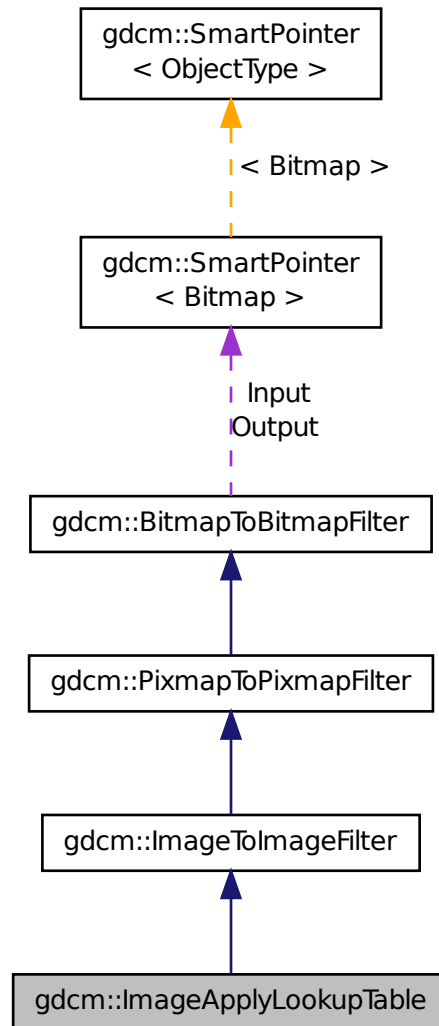
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for `gdcm::ImageApplyLookupTable`:



Public Member Functions

- `ImageApplyLookupTable ()`
- `~ImageApplyLookupTable ()`
- `bool Apply ()`

Apply.

25.128.1 Detailed Description

ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

25.128.2 Constructor & Destructor Documentation

25.128.2.1 `gdcm::ImageApplyLookupTable::ImageApplyLookupTable ()` `[inline]`

25.128.2.2 `gdcm::ImageApplyLookupTable::~~ImageApplyLookupTable ()` `[inline]`

25.128.3 Member Function Documentation

25.128.3.1 `bool gdcm::ImageApplyLookupTable::Apply ()`

Apply.

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

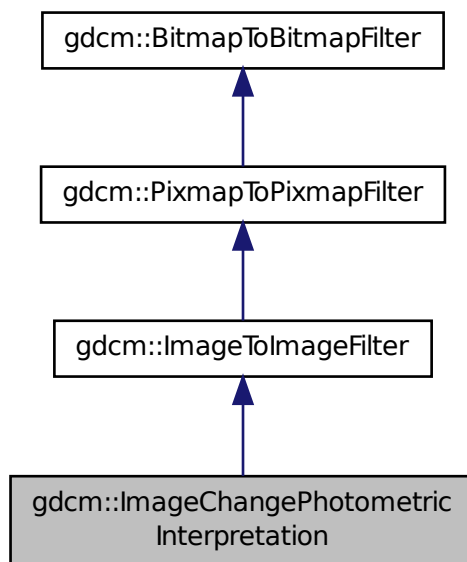
- `gdcmImageApplyLookupTable.h`

25.129 gdcm::ImageChangePhotometricInterpretation Class Reference

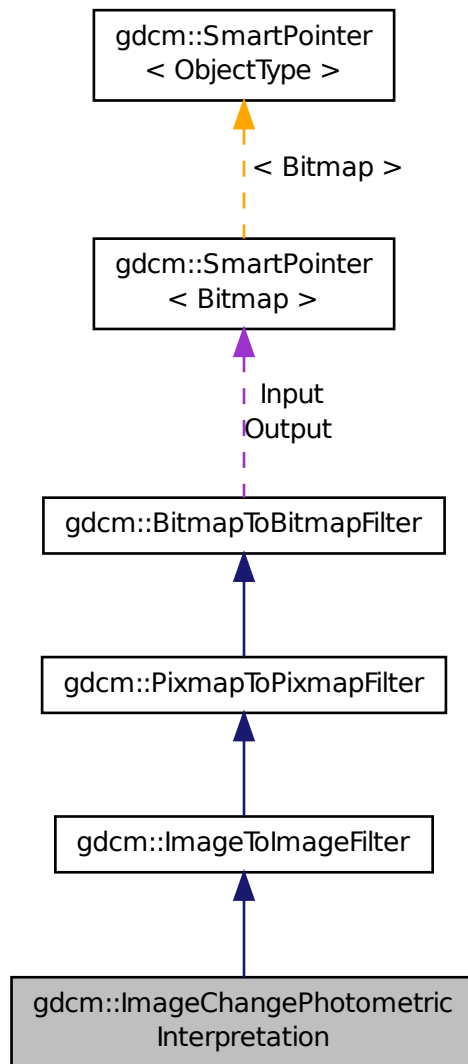
ImageChangePhotometricInterpretation class Class to change the Photometric Interpretation of an input DICOM.

```
#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 ()`

25.129.1 Detailed Description

ImageChangePhotometricInterpretation class Class to change the Photometric Interpretation of an input DICOM.

25.129.2 Constructor & Destructor Documentation

25.129.2.1 `gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ()` `[inline]`

25.129.2.2 `gdcm::ImageChangePhotometricInterpretation::~~ImageChangePhotometricInterpretation ()` `[inline]`

25.129.3 Member Function Documentation

25.129.3.1 `bool gdcm::ImageChangePhotometricInterpretation::Change ()`

Change.

25.129.3.2 `bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ()` `[protected]`

25.129.3.3 `const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation ()`
`const` `[inline]`

25.129.3.4 `template<typename T > void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (T ybr[3], const T rgb[3])`
`[static]`

colorspace conversion (based on CCIR Recommendation 601-2)

25.129.3.5 `void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (PhotometricInterpretation const & pi)` `[inline]`

Set/Get requested PhotometricInterpretation.

25.129.3.6 `template<typename T > void gdcm::ImageChangePhotometricInterpretation::YBR2RGB (T rgb[3], const T ybr[3])`
`[static]`

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

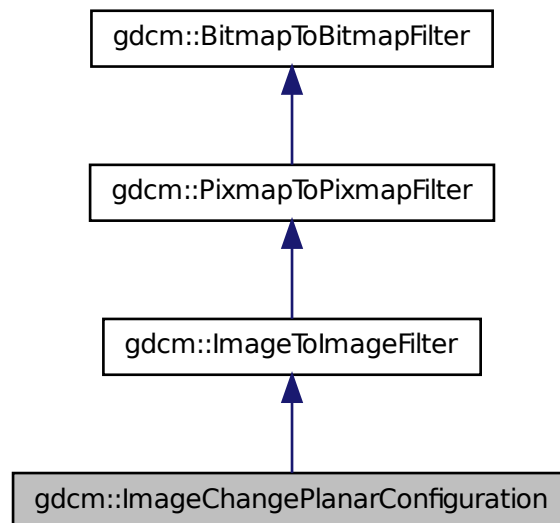
- `gdcmImageChangePhotometricInterpretation.h`

25.130 gdcm::ImageChangePlanarConfiguration Class Reference

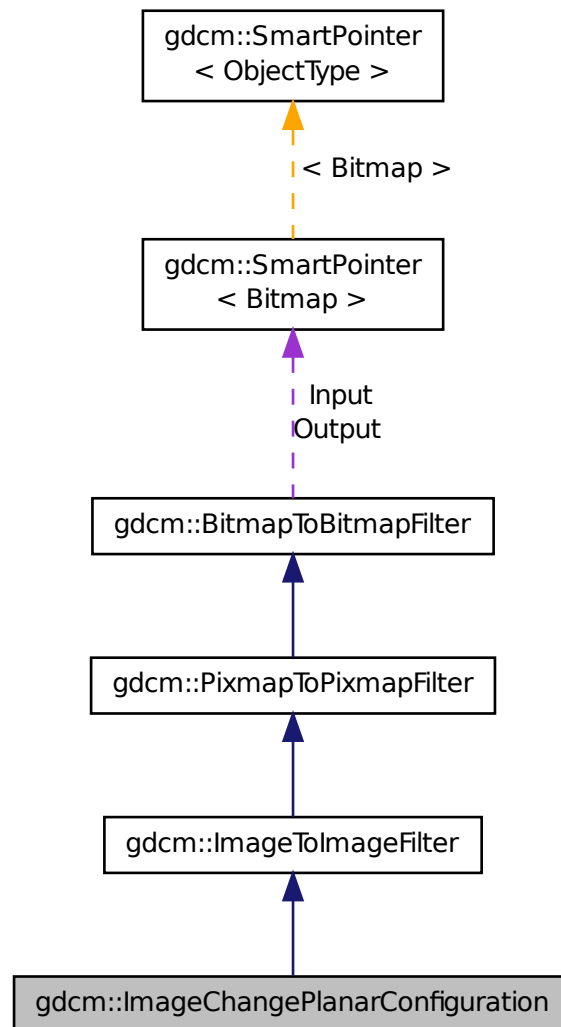
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0.

```
#include <gdcmImageChangePlanarConfiguration.h>
```

Inheritance diagram for gdcm::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)`

25.130.1 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.

25.130.2 Constructor & Destructor Documentation

25.130.2.1 `gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ()` `[inline]`

25.130.2.2 `gdcm::ImageChangePlanarConfiguration::~~ImageChangePlanarConfiguration ()` `[inline]`

25.130.3 Member Function Documentation

25.130.3.1 `bool gdcm::ImageChangePlanarConfiguration::Change ()`

Change.

25.130.3.2 `unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration () const` `[inline]`

25.130.3.3 `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

25.130.3.4 `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

25.130.3.5 `void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (unsigned int pc)` `[inline]`

Set/Get requested PlanarConfiguration.

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

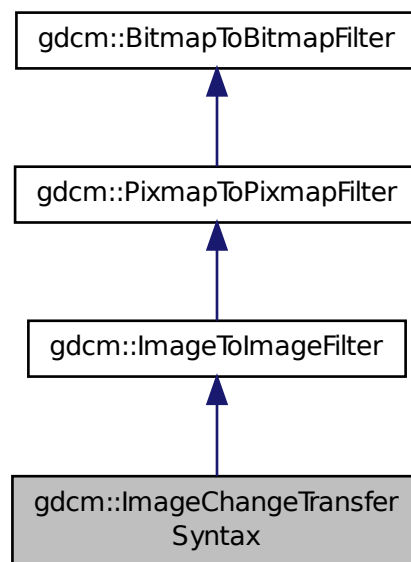
- `gdcmImageChangePlanarConfiguration.h`

25.131 gdcm::ImageChangeTransferSyntax Class Reference

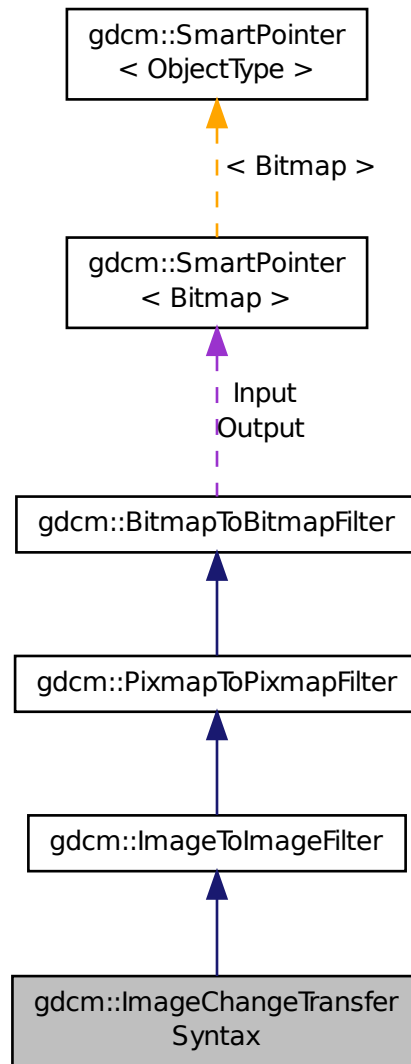
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.

```
#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)

25.131.1 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.

25.131.2 Constructor & Destructor Documentation

25.131.2.1 `gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ()` `[inline]`

25.131.2.2 `gdcm::ImageChangeTransferSyntax::~~ImageChangeTransferSyntax ()` `[inline]`

25.131.3 Member Function Documentation

25.131.3.1 `bool gdcm::ImageChangeTransferSyntax::Change ()`

Change.

Examples:

CompressImage.cxx.

25.131.3.2 `const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax () const` `[inline]`

Get Transfer Syntax.

25.131.3.3 void gdcm::ImageChangeTransferSyntax::SetCompressIconImage (bool *b*) [inline]

Decide whether or not to also compress the Icon Image using the same Transfer Syntax Default is to simply decompress icon image

25.131.3.4 void gdcm::ImageChangeTransferSyntax::SetForce (bool *f*) [inline]

When target Transfer Syntax is identical to input target syntax, no operation is actually done This is an issue when someone wants to recompress using GDCM internal implementation a JPEG (for example) image

25.131.3.5 void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (const TransferSyntax & *ts*) [inline]

Set target Transfer Syntax.

Examples:

CompressImage.cxx.

25.131.3.6 void gdcm::ImageChangeTransferSyntax::SetUserCodec (ImageCodec * *ic*) [inline]

Allow user to specify exactly which codec to use. this is needed to specify special qualities or compression option.

Warning

is 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)

25.131.3.7 bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

25.131.3.8 bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

25.131.3.9 bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

25.131.3.10 bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

25.131.3.11 bool gdcm::ImageChangeTransferSyntax::TryRLECodec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

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

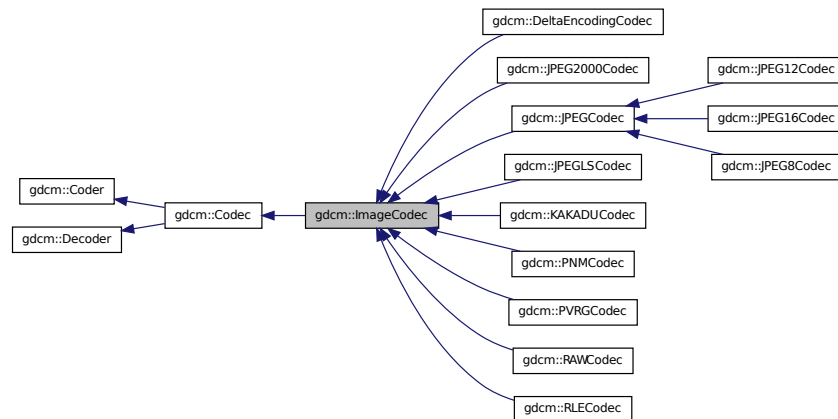
- gdcmImageChangeTransferSyntax.h

25.132 gdcm::ImageCodec Class Reference

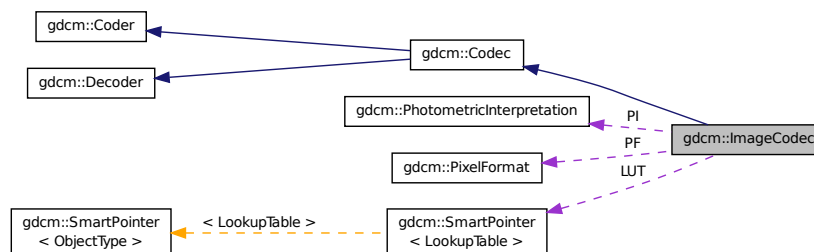
ImageCodec.

```
#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)
- 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)
- 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

- bool Decode (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 IsValid (PhotometricInterpretation const &pi)

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 ImageChangePhotometricInterpretation

25.132.1 Detailed Description

ImageCodec.

Note

Main codec, this is a central place for all implementation

25.132.2 Member Typedef Documentation

25.132.2.1 `typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr` `[protected]`

25.132.3 Constructor & Destructor Documentation

25.132.3.1 `gdcm::ImageCodec::ImageCodec ()`

25.132.3.2 `gdcm::ImageCodec::~~ImageCodec ()`

25.132.4 Member Function Documentation

25.132.4.1 `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::PVRGCodec`, `gdcm::JPEG2000Codec`, `gdcm::RLECodec`, `gdcm::JPEG-LSCoec`, `gdcm::PNMCodec`, `gdcm::KAKADUCodec`, and `gdcm::RAWCodec`.

25.132.4.2 `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::PVRGCodec`, `gdcm::RLECodec`, `gdcm::JPEG2000Codec`, `gdcm::JPEG-LSCoec`, `gdcm::PNMCodec`, `gdcm::RAWCodec`, and `gdcm::KAKADUCodec`.

25.132.4.3 `bool gdcm::ImageCodec::Decode (DataElement const & is, DataElement & os)` `[virtual]`

Decode.

Reimplemented from `gdcm::Decoder`.

Reimplemented in `gdcm::JPEGCodec`, `gdcm::PVRGCodec`, `gdcm::JPEGLSCoec`, `gdcm::JPEG2000Codec`, `gdcm::RLECodec`, `gdcm::DeltaEncodingCodec`, `gdcm::KAKADUCodec`, and `gdcm::RAWCodec`.

25.132.4.4 `bool gdcm::ImageCodec::Decode (std::istream & is, std::ostream & os)` `[protected]`, `[virtual]`

Reimplemented from `gdcm::Decoder`.

Reimplemented in `gdcm::JPEGCodec`, `gdcm::JPEG2000Codec`, `gdcm::RLECodec`, `gdcm::RAWCodec`, `gdcm::Delta-EncodingCodec`, `gdcm::JPEG12Codec`, `gdcm::JPEG16Codec`, and `gdcm::JPEG8Codec`.

25.132.4.5 `bool gdcm::ImageCodec::DoByteSwap (std::istream & is, std::ostream & os)` `[protected]`

25.132.4.6 `bool gdcm::ImageCodec::DoInvertMonochrome (std::istream & is, std::ostream & os)` `[protected]`

25.132.4.7 `bool gdcm::ImageCodec::DoOverlayCleanup (std::istream & is, std::ostream & os)` `[protected]`

25.132.4.8 `bool gdcm::ImageCodec::DoPaddedCompositePixelCode (std::istream & is, std::ostream & os)` `[protected]`

25.132.4.9 `bool gdcm::ImageCodec::DoPlanarConfiguration (std::istream & is, std::ostream & os)` `[protected]`

25.132.4.10 `bool gdcm::ImageCodec::DoSimpleCopy (std::istream & is, std::ostream & os)` `[protected]`

25.132.4.11 `bool gdcm::ImageCodec::DoYBR (std::istream & is, std::ostream & os)` `[protected]`

25.132.4.12 `const unsigned int* gdcm::ImageCodec::GetDimensions () const` `[inline]`

25.132.4.13 `virtual bool gdcm::ImageCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` `[virtual]`

Reimplemented in `gdcm::JPEGCodec`, `gdcm::JPEGLSCodec`, `gdcm::RLECodec`, `gdcm::JPEG2000Codec`, `gdcm::PN-MCodec`, `gdcm::JPEG12Codec`, `gdcm::JPEG16Codec`, `gdcm::JPEG8Codec`, and `gdcm::RAWCodec`.

25.132.4.14 `bool gdcm::ImageCodec::GetLossyFlag () const`

25.132.4.15 `const LookupTable& gdcm::ImageCodec::GetLUT () const` `[inline]`

25.132.4.16 `bool gdcm::ImageCodec::GetNeedByteSwap () const` `[inline]`

25.132.4.17 `unsigned int gdcm::ImageCodec::GetNumberOfDimensions () const`

25.132.4.18 `const PhotometricInterpretation& gdcm::ImageCodec::GetPhotometricInterpretation () const`

25.132.4.19 `PixelFormat& gdcm::ImageCodec::GetPixelFormat ()` `[inline]`

Examples:

`GetJPEGSamplePrecision.cxx`.

25.132.4.20 `const PixelFormat& gdcm::ImageCodec::GetPixelFormat () const` `[inline]`

25.132.4.21 `unsigned int gdcm::ImageCodec::GetPlanarConfiguration () const` `[inline]`

25.132.4.22 `bool gdcm::ImageCodec::IsLossy () const`

25.132.4.23 `virtual bool gdcM::ImageCodec::IsValid (PhotometricInterpretation const & pi)` [protected],
[virtual]

Reimplemented in `gdcM::JPEGCodec`.

25.132.4.24 `void gdcM::ImageCodec::SetDimensions (const unsigned int * d)` [inline]

Examples:

ExtractIconFromFile.cxx.

25.132.4.25 `void gdcM::ImageCodec::SetDimensions (const std::vector< unsigned int > & d)` [inline]

25.132.4.26 `void gdcM::ImageCodec::SetLossyFlag (bool l)`

25.132.4.27 `void gdcM::ImageCodec::SetLUT (LookupTable const & lut)` [inline]

Examples:

ExtractIconFromFile.cxx.

25.132.4.28 `void gdcM::ImageCodec::SetNeedByteSwap (bool b)` [inline]

25.132.4.29 `void gdcM::ImageCodec::SetNeedOverlayCleanup (bool b)` [inline]

25.132.4.30 `void gdcM::ImageCodec::SetNumberOfDimensions (unsigned int dim)`

25.132.4.31 `void gdcM::ImageCodec::SetPhotometricInterpretation (PhotometricInterpretation const & pi)`

Examples:

ExtractIconFromFile.cxx.

25.132.4.32 `virtual void gdcM::ImageCodec::SetPixelFormat (PixelFormat const & pf)` [inline],[virtual]

Reimplemented in `gdcM::JPEGCodec`.

Examples:

ExtractIconFromFile.cxx.

25.132.4.33 `void gdcM::ImageCodec::SetPlanarConfiguration (unsigned int pc)` [inline]

25.132.5 Friends And Related Function Documentation

25.132.5.1 `friend class ImageChangePhotometricInterpretation` [friend]

25.132.6 Member Data Documentation

- 25.132.6.1 unsigned int gdcm::ImageCodec::Dimensions[3] [protected]
- 25.132.6.2 bool gdcm::ImageCodec::LossyFlag [protected]
- 25.132.6.3 LUTPtr gdcm::ImageCodec::LUT [protected]
- 25.132.6.4 bool gdcm::ImageCodec::NeedByteSwap [protected]
- 25.132.6.5 bool gdcm::ImageCodec::NeedOverlayCleanup [protected]
- 25.132.6.6 unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]
- 25.132.6.7 PixelFormat gdcm::ImageCodec::PF [protected]
- 25.132.6.8 PhotometricInterpretation gdcm::ImageCodec::PI [protected]
- 25.132.6.9 unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]
- 25.132.6.10 bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]
- 25.132.6.11 bool gdcm::ImageCodec::RequestPlanarConfiguration [protected]

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

- gdcmImageCodec.h

25.133 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)

25.133.1 Detailed Description

Image Converter.

Note

This is the class used to convert from one gdcm::Image to another. This is typically used to convert let say YBR JPEG compressed gdcm::Image to a RAW RGB gdcm::Image. So that the buffer can be directly pass to third party application. This filter is application level and not integrated directly in GDCM

25.133.2 Constructor & Destructor Documentation

25.133.2.1 `gdcm::ImageConverter::ImageConverter ()`

25.133.2.2 `gdcm::ImageConverter::~~ImageConverter ()`

25.133.3 Member Function Documentation

25.133.3.1 `void gdcm::ImageConverter::Convert ()`

25.133.3.2 `const Image& gdcm::ImageConverter::GetOutput () const`

25.133.3.3 `void gdcm::ImageConverter::SetInput (Image const & input)`

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

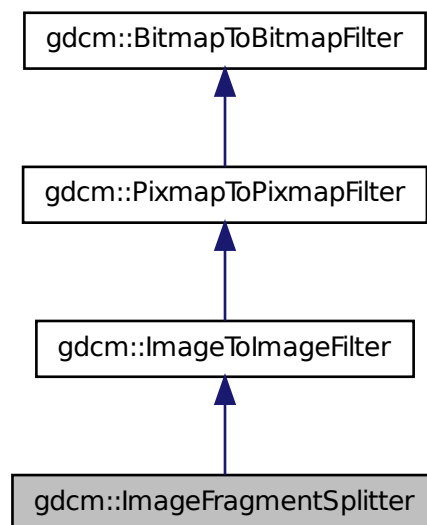
- `gdcmImageConverter.h`

25.134 `gdcm::ImageFragmentSplitter` Class Reference

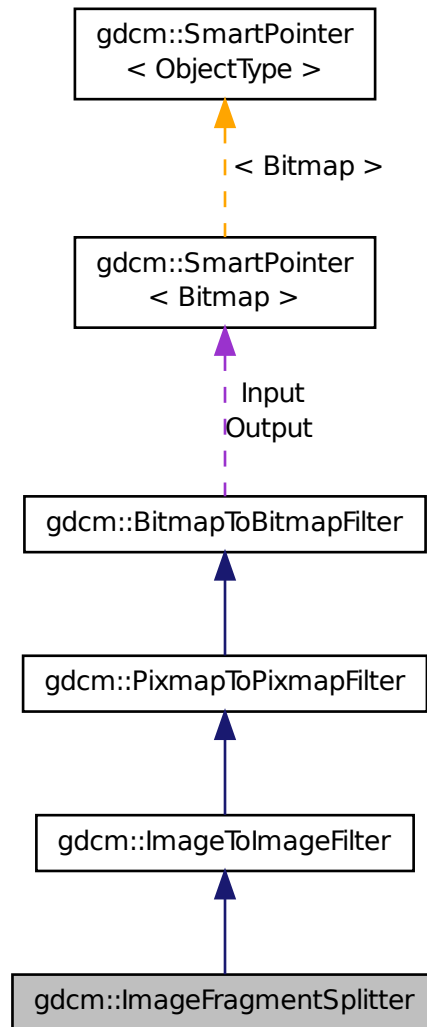
`ImageFragmentSplitter` class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

```
#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.

25.134.1 Detailed Description

ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

25.134.2 Constructor & Destructor Documentation

25.134.2.1 `gdcm::ImageFragmentSplitter::ImageFragmentSplitter ()` `[inline]`

25.134.2.2 `gdcm::ImageFragmentSplitter::~~ImageFragmentSplitter ()` `[inline]`

25.134.3 Member Function Documentation

25.134.3.1 `unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax () const` `[inline]`

25.134.3.2 `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 !

25.134.3.3 `void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (unsigned int fragsize)`

FragmentSizeMax needs to be an even number.

25.134.3.4 `bool gdcm::ImageFragmentSplitter::Split ()`

Split.

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

- `gdcmImageFragmentSplitter.h`

25.135 gdcm::ImageHelper Class Reference

ImageHelper (internal class, not intended for user level)

```
#include <gdcmImageHelper.h>
```

Static Public Member Functions

- `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)`
- `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)

Moved from PixampReader to here. Generally used for photometric interpretation.

- 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 Image &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)

25.135.1 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.

25.135.2 Member Function Documentation

25.135.2.1 static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (const std::vector< double > & *imageposition*, std::vector< double > & *spacing*) [static]

DO NOT USE.

25.135.2.2 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.

25.135.2.3 `static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (DataSet const & ds, std::vector< double > & dircos) [static]`

25.135.2.4 `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

25.135.2.5 `static bool gdcm::ImageHelper::GetForcePixelSpacing () [static]`

25.135.2.6 `static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope () [static]`

25.135.2.7 `static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (File const & f) [static]`

25.135.2.8 `static std::vector<double> gdcm::ImageHelper::GetOriginValue (File const & f) [static]`

Set/Get Origin (IPP) from/to a file.

25.135.2.9 `static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (File const & f) [static]`

25.135.2.10 `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)

25.135.2.11 `static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (const File & f) [static]`

25.135.2.12 `static const ByteValue* gdcm::ImageHelper::GetPointerFromElement (Tag const & tag, File const & f) [static]`

Moved from PixampReader to here. Generally used for photometric interpretation.

25.135.2.13 `static std::vector<double> gdcm::ImageHelper::GetRescaleInterceptSlopeValue (File const & f) [static]`

Set/Get shift/scale from/to a file

Warning

this function reads/sets the Slope/Intercept in appropriate class storage, but also Grid Scaling in RT Dose Storage
Can't take a dataset because the mediastorage of the file must be known

25.135.2.14 `static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (MediaStorage const & ms) [static], [protected]`

25.135.2.15 `static std::vector<double> gdcm::ImageHelper::GetSpacingValue (File const & f) [static]`

Set/Get Spacing from/to a File.

25.135.2.16 `static Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (MediaStorage const & ms) [static], [protected]`

25.135.2.17 `static void gdcm::ImageHelper::SetDimensionsValue (File & f, const Image & img) [static]`

25.135.2.18 `static void gdcm::ImageHelper::SetDirectionCosinesValue (DataSet & ds, const std::vector< double > & dircos) [static]`

Set Direction Cosines (IOP) from/to a file When IOD does not defines what is IOP (eg. typically Secondary Capture Image Storage) this call will simply remove the IOP attribute. Else in case of MR/CT image storage, this call will properly lookup the correct attribute to store the IOP.

25.135.2.19 `static void gdcm::ImageHelper::SetForcePixelSpacing (bool) [static]`

GDCM 1.x compatibility issue: When using ReWrite an MR Image Storage would be rewritten as Secondary Capture Object while still having a Pixel Spacing tag (0028,0030). If you have deal with those files, use this very special flag to handle them Unless explicitey set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel Spacing of the Image

25.135.2.20 `static void gdcm::ImageHelper::SetForceRescaleInterceptSlope (bool) [static]`

GDCM 1.x compatibility issue: when using ReWrite an MR Image Storage would be rewritten with a Rescale Slope/- Intercept while the standard would prohibit this (Philips Medical System is still doing that) Unless explicitey set elsewhere by the standard, it will use value from 0028,1052 / 0028,1053 for the Rescale Slope & Rescale Intercept values

25.135.2.21 `static void gdcm::ImageHelper::SetOriginValue (DataSet & ds, const Image & img) [static]`

25.135.2.22 `static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (File & f, const Image & img) [static]`

25.135.2.23 `static void gdcm::ImageHelper::SetSpacingValue (DataSet & ds, const std::vector< double > & spacing) [static]`

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

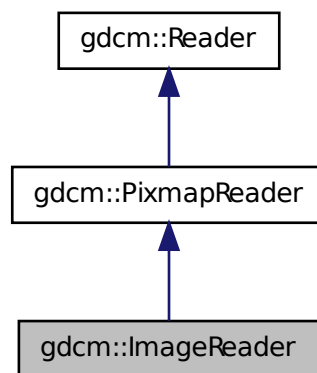
- `gdcmImageHelper.h`

25.136 gdcm::ImageReader Class Reference

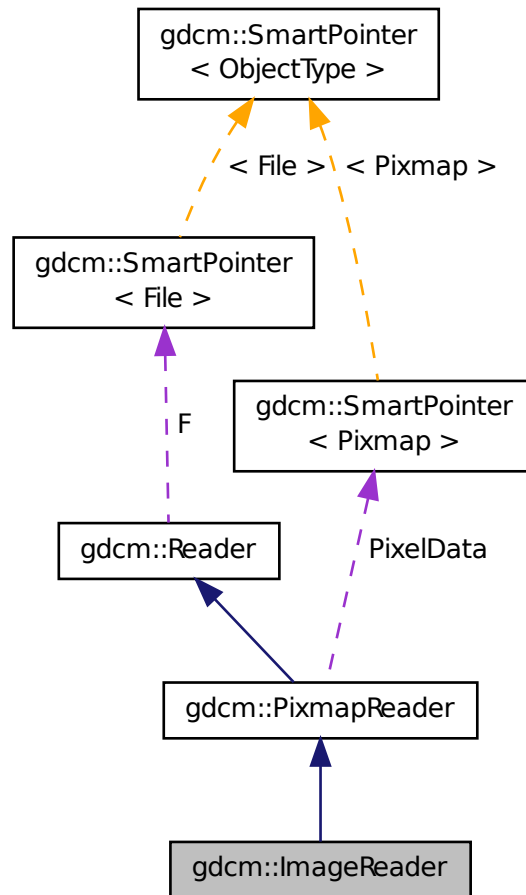
ImageReader.

```
#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

25.136.1 Detailed Description

ImageReader.

Note

its role is to convert the DICOM DataSet into a `gdcm::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`.

25.136.2 Constructor & Destructor Documentation

25.136.2.1 `gdcm::ImageReader::ImageReader ()`

25.136.2.2 `virtual gdcm::ImageReader::~~ImageReader ()` `[virtual]`

25.136.3 Member Function Documentation

25.136.3.1 `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`.

25.136.3.2 `Image& gdcm::ImageReader::GetImage ()`

25.136.3.3 `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`.

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.

25.136.3.4 `bool gdcm::ImageReader::ReadACRNEMAImage () [protected],[virtual]`

Reimplemented from `gdcm::PixmapReader`.

25.136.3.5 `bool gdcm::ImageReader::ReadImage (MediaStorage const & ms) [protected],[virtual]`

Reimplemented from `gdcm::PixmapReader`.

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

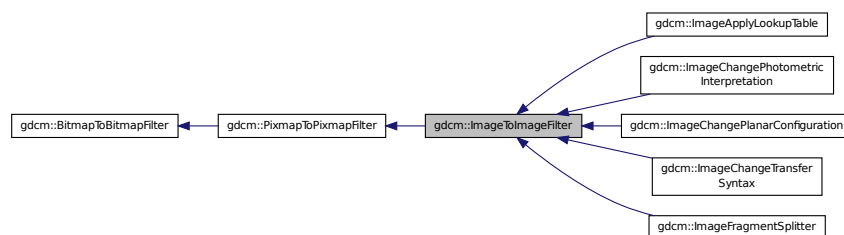
- `gdcmImageReader.h`

25.137 gdcm::ImageToImageFilter Class Reference

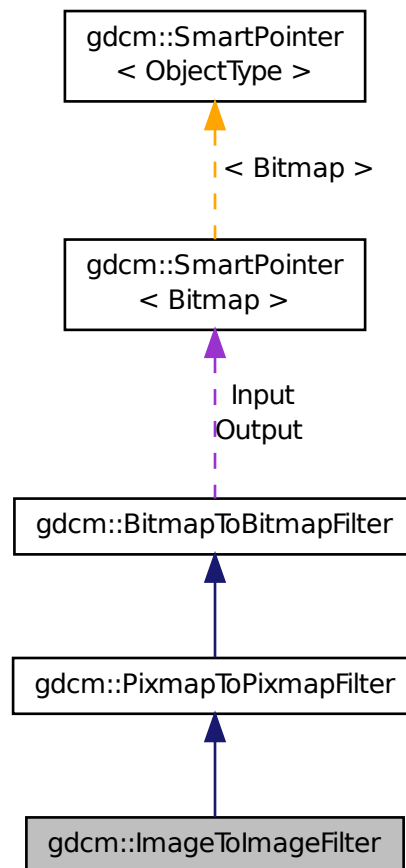
ImageToImageFilter class Super class for all filter taking an image and producing an output image.

```
#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.

25.137.1 Detailed Description

`ImageToImageFilter` class Super class for all filter taking an image and producing an output image.

25.137.2 Constructor & Destructor Documentation

25.137.2.1 `gdcm::ImageToImageFilter::ImageToImageFilter ()`

25.137.2.2 `gdcm::ImageToImageFilter::~~ImageToImageFilter ()` `[inline]`

25.137.3 Member Function Documentation

25.137.3.1 `Image& gdcm::ImageToImageFilter::GetInput ()`

Reimplemented from `gdcm::PixmapToPixmapFilter`.

25.137.3.2 `const Image& gdcm::ImageToImageFilter::GetOutput () const`

Get Output image.

Reimplemented from `gdcm::PixmapToPixmapFilter`.

Examples:

`CompressImage.cxx`.

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

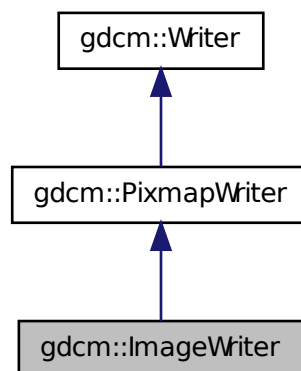
- `gdcmImageToImageFilter.h`

25.138 gdcm::ImageWriter Class Reference

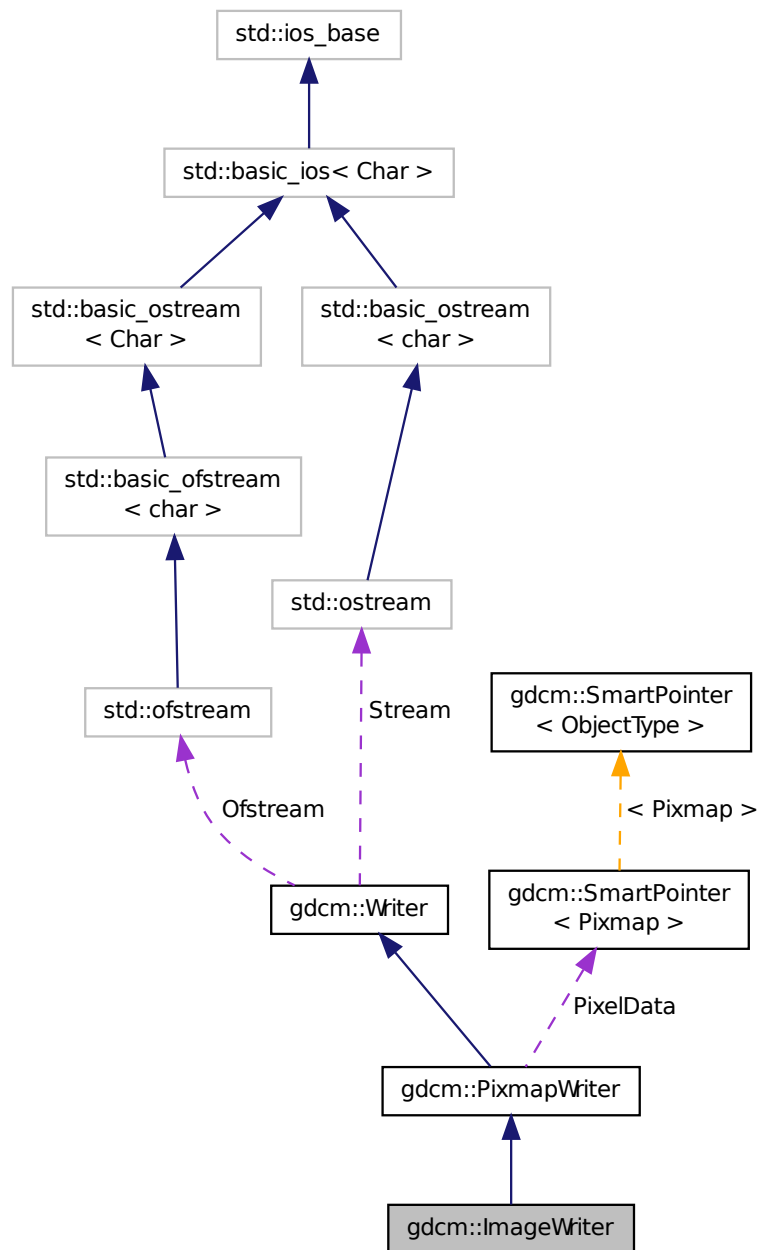
ImageWriter.

```
#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

25.138.1 Detailed Description

ImageWriter.

Examples:

CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, and MergeTwoFiles.cxx.

25.138.2 Constructor & Destructor Documentation

25.138.2.1 `gdcm::ImageWriter::ImageWriter ()`

25.138.2.2 `gdcm::ImageWriter::~~ImageWriter ()`

25.138.3 Member Function Documentation

25.138.3.1 `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.

25.138.3.2 `Image& gdcm::ImageWriter::GetImage ()` `[inline],[virtual]`

Reimplemented from `gdcm::PixmapWriter`.

25.138.3.3 `bool gdcm::ImageWriter::Write ()` `[virtual]`

Write.

Reimplemented from `gdcm::PixmapWriter`.

Examples:

CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, and MergeTwoFiles.cxx.

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

- `gdcmImageWriter.h`

25.139 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 ()
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.139.1 Detailed Description

ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

25.139.2 Constructor & Destructor Documentation

25.139.2.1 gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ()

25.139.3 Member Function Documentation

25.139.3.1 std::istream& gdcm::network::ImplementationClassUIDSub::Read (std::istream & *is*)

25.139.3.2 size_t gdcm::network::ImplementationClassUIDSub::Size () const

25.139.3.3 const std::ostream& gdcm::network::ImplementationClassUIDSub::Write (std::ostream & *os*) const

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

- gdcmImplementationClassUIDSub.h

25.140 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

25.140.1 Detailed Description

ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

25.140.2 Constructor & Destructor Documentation

25.140.2.1 `gdcm::network::ImplementationUIDSub::ImplementationUIDSub ()`

25.140.3 Member Function Documentation

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

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

- `gdcmImplementationUIDSub.h`

25.141 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 ()`
- `std::istream & Read (std::istream &is)`
- `size_t Size () const`
- `const std::ostream & Write (std::ostream &os) const`

25.141.1 Detailed Description

ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

25.141.2 Constructor & Destructor Documentation

25.141.2.1 `gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ()`

25.141.3 Member Function Documentation

25.141.3.1 `std::istream& gdcm::network::ImplementationVersionNameSub::Read (std::istream & is)`

25.141.3.2 `size_t gdcm::network::ImplementationVersionNameSub::Size () const`

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

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

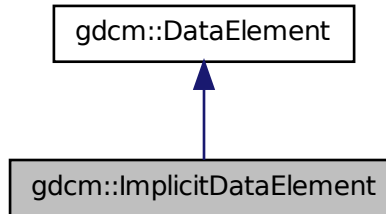
- `gdcmImplementationVersionNameSub.h`

25.142 gdcm::ImplicitDataElement Class Reference

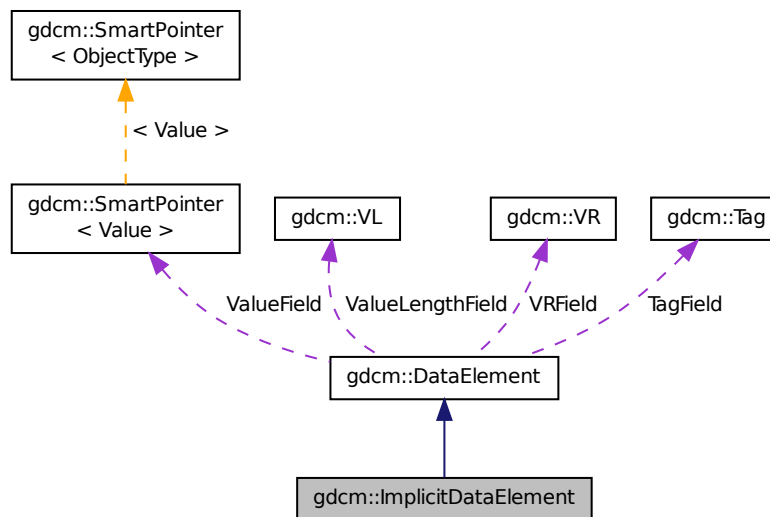
Class to represent an *Implicit VR* Data Element.

```
#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)`
- `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

25.142.1 Detailed Description

Class to represent an *Implicit VR* Data Element.

Note

bla

Examples:

ReadExplicitLengthSQIVR.cxx.

25.142.2 Member Function Documentation

25.142.2.1 VL gdcm::ImplicitDataElement::GetLength () const

Reimplemented from `gdcm::DataElement`.

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

Reimplemented from `gdcm::DataElement`.

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

25.142.2.4 `template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValue (std::istream & is)`

25.142.2.5 `template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

Reimplemented from `gdcm::DataElement`.

25.142.2.6 `template<typename TSwap > const std::ostream& gdcm::ImplicitDataElement::Write (std::ostream & os) const`

Reimplemented from `gdcm::DataElement`.

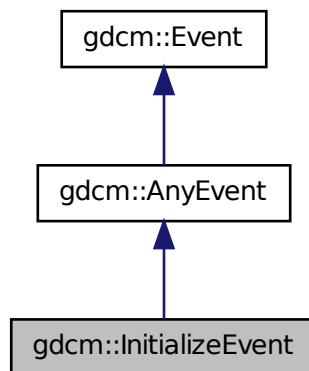
The documentation for this class was generated from the following file:

- `gdcmImplicitDataElement.h`

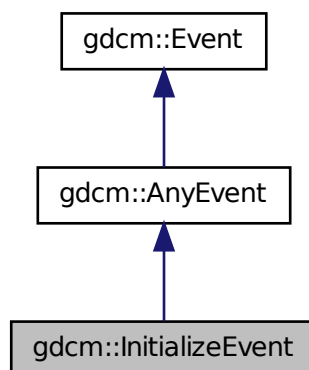
25.143 gdcm::InitializeEvent Class Reference

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

Inheritance diagram for gdcm::InitializeEvent:



Collaboration diagram for gdcm::InitializeEvent:



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

- gdcmEvent.h

25.144 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)

25.144.1 Detailed Description

Class for representing a IOD.

Note

bla

See also

Dict

Examples:

TraverseModules.cxx.

25.144.2 Member Typedef Documentation

25.144.2.1 typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry

25.144.2.2 typedef MapIODEntry::size_type gdcm::IOD::SizeType

25.144.3 Constructor & Destructor Documentation

25.144.3.1 gdcm::IOD::IOD () [inline]

25.144.4 Member Function Documentation

25.144.4.1 `void gdcmlOD::AddIODEntry (const IODEntry & iode)` `[inline]`

25.144.4.2 `void gdcmlOD::Clear ()` `[inline]`

25.144.4.3 `const IODEntry& gdcmlOD::GetIODEntry (SizeType idx) const` `[inline]`

Examples:

TraverseModules.cxx.

25.144.4.4 `SizeType gdcmlOD::GetNumberOfIODs () const` `[inline]`

Examples:

TraverseModules.cxx.

25.144.4.5 `Type gdcmlOD::GetTypeFromTag (const Defs & defs, const Tag & tag) const`

25.144.5 Friends And Related Function Documentation

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

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

- gdcmlOD.h

25.145 gdcmlODEntry Class Reference

Class for representing a IODEntry.

```
#include <gdcmlODEntry.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)`

25.145.1 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
- 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.

25.145.2 Constructor & Destructor Documentation

25.145.2.1 `gdcm::IODEntry::IODEntry (const char * name = " ", const char * ref = " ", const char * usag = " ") [inline]`

25.145.3 Member Function Documentation

25.145.3.1 `const char* gdcm::IODEntry::GetIE () const [inline]`

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

25.145.3.3 `const char* gdcm::IODEntry::GetRef () const [inline]`

Examples:

TraverseModules.cxx.

25.145.3.4 `const char* gdcml::IODEntry::GetUsage () const` [inline]

25.145.3.5 `Usage::UsageType gdcml::IODEntry::GetUsageType () const`

25.145.3.6 `void gdcml::IODEntry::SetIE (const char * ie)` [inline]

25.145.3.7 `void gdcml::IODEntry::SetName (const char * name)` [inline]

25.145.3.8 `void gdcml::IODEntry::SetRef (const char * ref)` [inline]

25.145.3.9 `void gdcml::IODEntry::SetUsage (const char * usag)` [inline]

25.145.4 Friends And Related Function Documentation

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

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

- `gdcmlIODEntry.h`

25.146 gdcml::IODs Class Reference

Class for representing a IODs.

```
#include <gdcmlIODs.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)`

25.146.1 Detailed Description

Class for representing a IODs.

Note

bla

See also

IOD

Examples:

TraverseModules.cxx.

25.146.2 Member Typedef Documentation

25.146.2.1 `typedef std::map<IODName, IOD> gdcm::IODs::IODMapType`

25.146.2.2 `typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator`

25.146.2.3 `typedef std::string gdcm::IODs::IODName`

25.146.3 Constructor & Destructor Documentation

25.146.3.1 `gdcm::IODs::IODs ()` `[inline]`

25.146.4 Member Function Documentation

25.146.4.1 `void gdcm::IODs::AddIOD (const char * name, const IOD & module)` `[inline]`

25.146.4.2 `IODMapTypeConstIterator gdcm::IODs::Begin () const` `[inline]`

25.146.4.3 `void gdcm::IODs::Clear ()` `[inline]`

25.146.4.4 `IODMapTypeConstIterator gdcm::IODs::End () const` `[inline]`

25.146.4.5 `const IOD& gdcm::IODs::GetIOD (const char * name) const` `[inline]`

25.146.5 Friends And Related Function Documentation

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

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

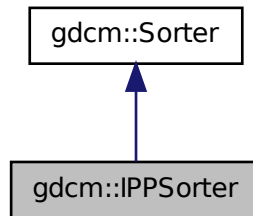
- gdcmIODs.h

25.147 gdcm::IPPSorter Class Reference

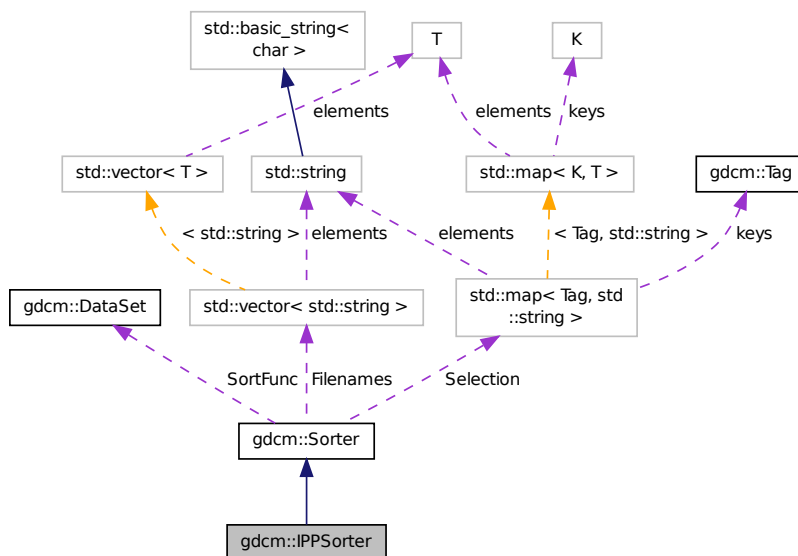
IPPSorter Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

```
#include <gdcmIPPSorter.h>
```

Inheritance diagram for `gdcm::IPPSorter`:



Collaboration diagram for `gdcm::IPPSorter`:



Public Member Functions

- `IPPSorter ()`
- `~IPPSorter ()`
- `double GetDirectionCosinesTolerance () const`
- `double GetZSpacing () const`
- `double GetZSpacingTolerance () const`
- `void SetComputeZSpacing (bool b)`

- void SetDirectionCosinesTolerance (double tol)
- void SetZSpacingTolerance (double tol)
- virtual bool Sort (std::vector< std::string > const &filenames)

Protected Attributes

- bool ComputeZSpacing
- double DirCosTolerance
- double ZSpacing
- double ZTolerance

Additional Inherited Members

25.147.1 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 constant spacing is: 1e-6mm)

For more information on Spacing, and how it is defined in DICOM, advanced users may refer to:

http://sourceforge.net/apps/mediawiki/gdcm/index.php?title=Imager_Pixel_Spacing

Bug There currently a couple of bug in this implementation:

- Frame Of Reference UID is not taken into account
- Gantry Tilt is not considered

Examples:

gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

25.147.2 Constructor & Destructor Documentation

25.147.2.1 `gdcm::IPPSorter::IPPSorter ()`

25.147.2.2 `gdcm::IPPSorter::~~IPPSorter ()`

25.147.3 Member Function Documentation

25.147.3.1 `double gdcm::IPPSorter::GetDirectionCosinesTolerance () const` `[inline]`

25.147.3.2 `double gdcm::IPPSorter::GetZSpacing () const` `[inline]`

Read-only function to provide access to the computed value for the Z-Spacing The ComputeZSpacing must have been set to true before execution of sort algorithm. Call this function *after* calling Sort(); Z-Spacing will be 0 on 2 occasions:

- Sorting simply failed, potentially duplicate IPP => ZSpacing = 0
- ZSpacing could not be computed (Z-Spacing is not constant, or ZTolerance is too low)

Examples:

gdcmorthoplanes.cxx, and reslicesphere.cxx.

25.147.3.3 `double gdcm::IPPSorter::GetZSpacingTolerance () const [inline]`

25.147.3.4 `void gdcm::IPPSorter::SetComputeZSpacing (bool b) [inline]`

Functions related to Z-Spacing computation Set to true when sort algorithm should also perform a regular Z-Spacing computation using the Image Position (Patient) Potential reason for failure:

1. ALL slices are taken into account, if one slice is missing then ZSpacing will be set to 0 since the spacing will not be found to be regular along the Series

Examples:

gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

25.147.3.5 `void gdcm::IPPSorter::SetDirectionCosinesTolerance (double tol) [inline]`

Sometimes IOP along a series is slightly changing for example: "0.999081\\0.0426953\\0.00369272\\-0.0419025\\0.-955059\\0.293439", "0.999081\\0.0426953\\0.00369275\\-0.0419025\\0.955059\\0.293439", "0.999081\\0.0426952\\0.-00369272\\-0.0419025\\0.955059\\0.293439", We need an API to define the tolerance which is allowed. Internally the cross vector of each direction cosines is computed. The tolerance then define the the distance in between 1. 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).

25.147.3.6 `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:

gdcmorthoplanes.cxx, and reslicesphere.cxx.

25.147.3.7 `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 (SetZSpacingTolerance, ...) Return value indicate if sorting could be achieved. Warning this does *NOT* imply that spacing is constant, 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:

gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

25.147.4 Member Data Documentation

25.147.4.1 `bool gdcm::IPPSorter::ComputeZSpacing` `[protected]`

25.147.4.2 `double gdcm::IPPSorter::DirCosTolerance` `[protected]`

25.147.4.3 `double gdcm::IPPSorter::ZSpacing` `[protected]`

25.147.4.4 `double gdcm::IPPSorter::ZTolerance` `[protected]`

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

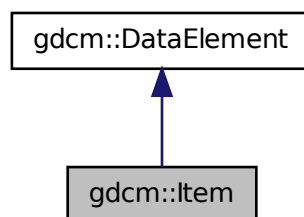
- `gdcmIPPSorter.h`

25.148 gdcm::Item Class Reference

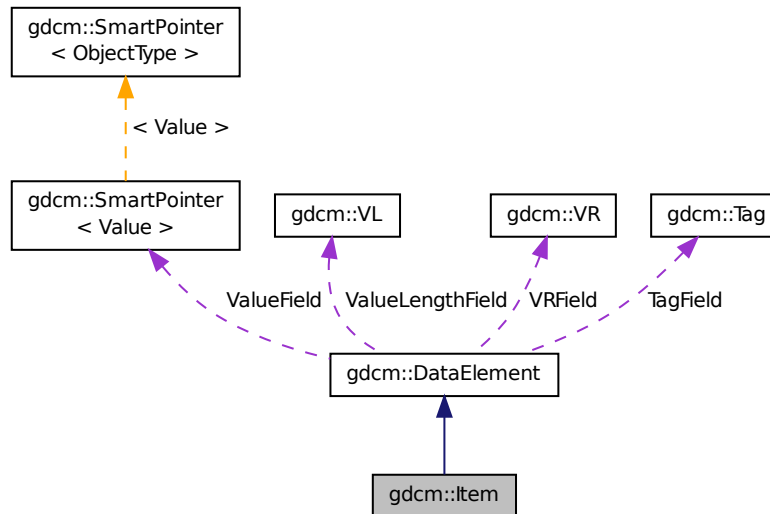
Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standart Data Element with a specific Data Element 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 `gdcm::Item`:



Collaboration diagram for `gdcM::Item`:



Public Member Functions

- `Item ()`
- `Item (Item const &val)`
- `void Clear ()`
Clear Data Element (make Value empty and invalidate Tag & VR)
- `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

25.148.1 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, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllIVR.cxx, GenFakeldentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, and LargeVRDSEExplicit.cxx.

25.148.2 Constructor & Destructor Documentation

25.148.2.1 `gdcm::Item::Item () [inline]`

25.148.2.2 `gdcm::Item::Item (Item const & val) [inline]`

25.148.3 Member Function Documentation

25.148.3.1 `void gdcm::Item::Clear () [inline]`

Clear Data Element (make Value empty and invalidate Tag & VR)

Reimplemented from `gdcm::DataElement`.

References `gdcm::DataElement::Clear()`.

Referenced by `gdcm::SequenceOfItems::Read()`.

25.148.3.2 `bool gdcm::Item::FindDataElement (const Tag & t) const [inline]`

Examples:

ReadAndDumpDICOMDIR.cxx.

25.148.3.3 `const DataElement& gdcm::Item::GetDataElement (const Tag & t) const [inline]`

Examples:

ReadAndDumpDICOMDIR.cxx.

25.148.3.4 `template<typename TDE > VL gdcm::Item::GetLength () const`

Reimplemented from `gdcm::DataElement`.

25.148.3.5 `const DataSet& gdcm::Item::GetNestedDataSet () const` `[inline]`

Examples:

ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.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()`.

25.148.3.6 `DataSet& gdcm::Item::GetNestedDataSet ()` `[inline]`

25.148.3.7 `void gdcm::Item::InsertDataElement (const DataElement & de)` `[inline]`

25.148.3.8 `template<typename TDE , typename TSwap > std::istream& gdcm::Item::Read (std::istream & is)` `[inline]`

Reimplemented from `gdcm::DataElement`.

References `gdcm::DataSet::Clear()`, `gdcmDebugMacro`, `gdcmErrorMacro`, `gdcmWarningMacro`, `gdcm::DataSet::IsEmpty()`, and `gdcm::SwapperDoOp::Swap()`.

Referenced by `gdcm::SequenceOfItems::Read()`.

25.148.3.9 `void gdcm::Item::SetNestedDataSet (const DataSet & nested)` `[inline]`

25.148.3.10 `template<typename TDE , typename TSwap > const std::ostream& gdcm::Item::Write (std::ostream & os) const` `[inline]`

Reimplemented from `gdcm::DataElement`.

References `gdcmWarningMacro`, `gdcm::VL::GetLength()`, `gdcm::VL::Write()`, and `gdcm::Tag::Write()`.

25.148.4 Friends And Related Function Documentation

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

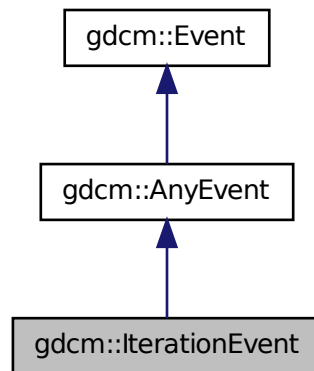
The documentation for this class was generated from the following file:

- `gdcmItem.h`

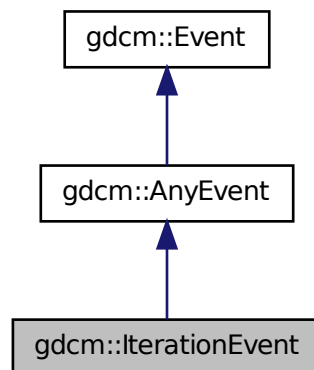
25.149 gdcm::IterationEvent Class Reference

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

Inheritance diagram for gdcm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



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

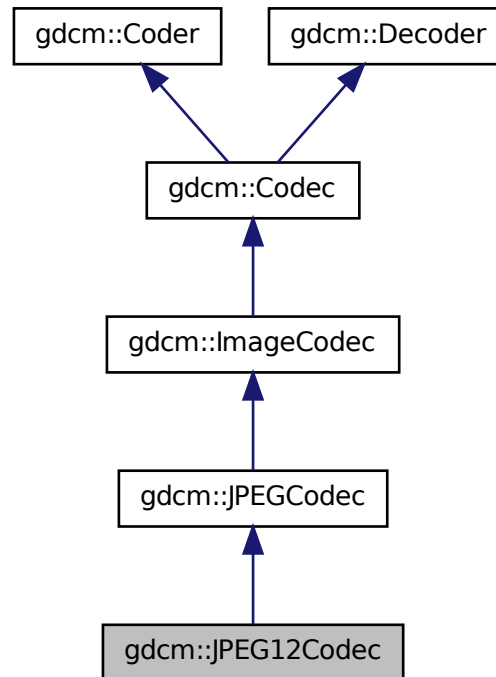
- gdcmEvent.h

25.150 gdcm::JPEG12Codec Class Reference

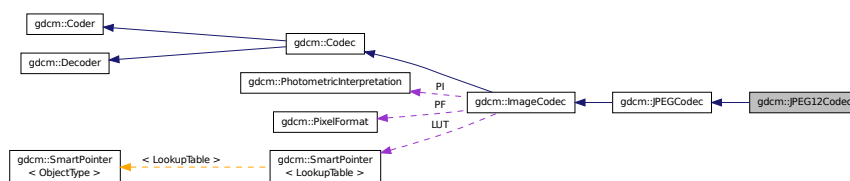
Class to do JPEG 12bits (lossy & lossless)

```
#include <gdcmJPEG12Codec.h>
```

Inheritance diagram for `gdcm::JPEG12Codec`:



Collaboration diagram for `gdcm::JPEG12Codec`:



Public Member Functions

- `JPEG12Codec ()`
- `~JPEG12Codec ()`
- `bool Decode (std::istream &is, std::ostream &os)`
- `bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)`
- `bool InternalCode (const char *input, unsigned long len, std::ostream &os)`

Additional Inherited Members

25.150.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

Note

internal class

25.150.2 Constructor & Destructor Documentation

25.150.2.1 `gdcm::JPEG12Codec::JPEG12Codec ()`

25.150.2.2 `gdcm::JPEG12Codec::~~JPEG12Codec ()`

25.150.3 Member Function Documentation

25.150.3.1 `bool gdcm::JPEG12Codec::Decode (std::istream & is, std::ostream & os)` [virtual]

Reimplemented from `gdcm::JPEGCodec`.

25.150.3.2 `bool gdcm::JPEG12Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from `gdcm::JPEGCodec`.

25.150.3.3 `bool gdcm::JPEG12Codec::InternalCode (const char * input, unsigned long len, std::ostream & os)` [virtual]

Reimplemented from `gdcm::Coder`.

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

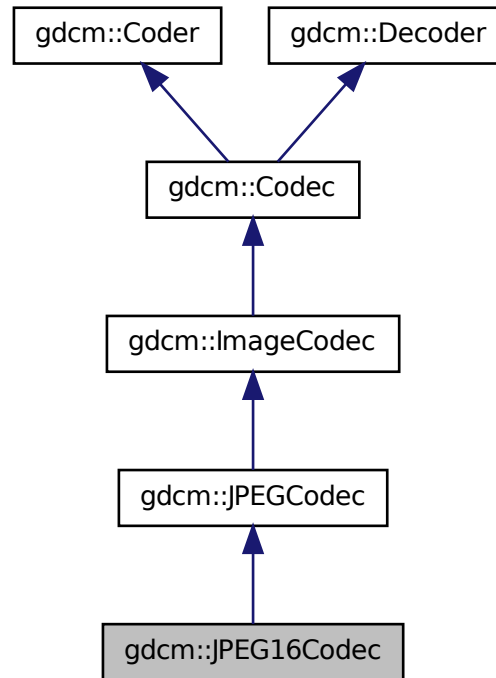
- `gdcmJPEG12Codec.h`

25.151 gdcm::JPEG16Codec Class Reference

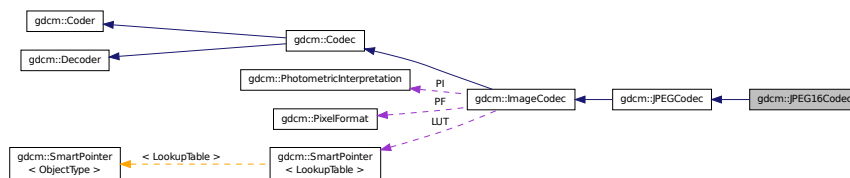
Class to do JPEG 16bits (lossless)

```
#include <gdcmJPEG16Codec.h>
```

Inheritance diagram for `gdcm::JPEG16Codec`:



Collaboration diagram for `gdcm::JPEG16Codec`:



Public Member Functions

- `JPEG16Codec ()`
- `~JPEG16Codec ()`
- `bool Decode (std::istream &is, std::ostream &os)`
- `bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)`
- `bool InternalCode (const char *input, unsigned long len, std::ostream &os)`

Additional Inherited Members

25.151.1 Detailed Description

Class to do JPEG 16bits (lossless)

Note

internal class

25.151.2 Constructor & Destructor Documentation

25.151.2.1 `gdcm::JPEG16Codec::JPEG16Codec ()`

25.151.2.2 `gdcm::JPEG16Codec::~~JPEG16Codec ()`

25.151.3 Member Function Documentation

25.151.3.1 `bool gdcm::JPEG16Codec::Decode (std::istream & is, std::ostream & os)` [virtual]

Reimplemented from `gdcm::JPEGCodec`.

25.151.3.2 `bool gdcm::JPEG16Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from `gdcm::JPEGCodec`.

25.151.3.3 `bool gdcm::JPEG16Codec::InternalCode (const char * input, unsigned long len, std::ostream & os)` [virtual]

Reimplemented from `gdcm::Coder`.

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

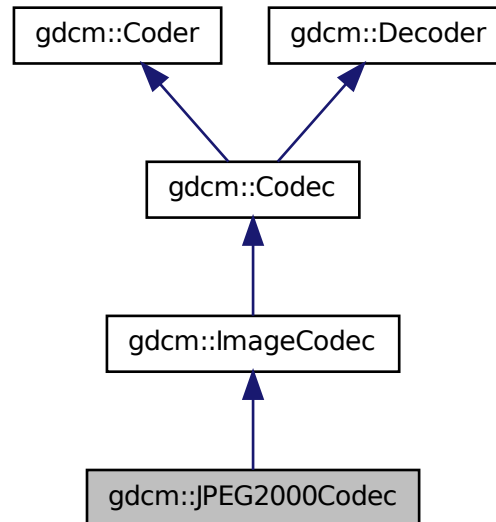
- `gdcmJPEG16Codec.h`

25.152 gdcm::JPEG2000Codec Class Reference

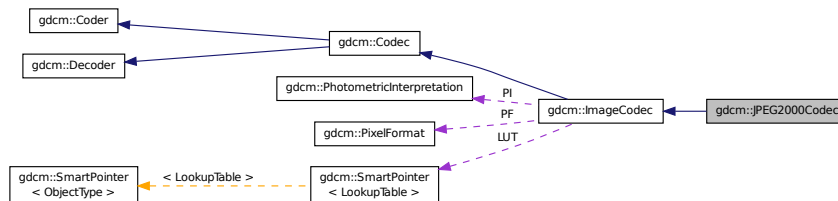
Class to do JPEG 2000.

```
#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)
- `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 Decode (std::istream &is, std::ostream &os)

Friends

- class Bitmap

Additional Inherited Members

25.152.1 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

25.152.2 Constructor & Destructor Documentation

25.152.2.1 gdcm::JPEG2000Codec::JPEG2000Codec ()

25.152.2.2 gdcm::JPEG2000Codec::~~JPEG2000Codec ()

25.152.3 Member Function Documentation

25.152.3.1 bool gdcm::JPEG2000Codec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.152.3.2 bool gdcm::JPEG2000Codec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.152.3.3 `bool gdcmm::JPEG2000Codec::Code (DataElement const & in_, DataElement & out_)` [virtual]

Code.

Reimplemented from `gdcmm::Coder`.

25.152.3.4 `bool gdcmm::JPEG2000Codec::Decode (DataElement const & is_, DataElement & os)` [virtual]

Decode.

Reimplemented from `gdcmm::ImageCodec`.

25.152.3.5 `bool gdcmm::JPEG2000Codec::Decode (std::istream & is, std::ostream & os)` [protected], [virtual]

Reimplemented from `gdcmm::ImageCodec`.

25.152.3.6 `virtual bool gdcmm::JPEG2000Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from `gdcmm::ImageCodec`.

25.152.3.7 `double gdcmm::JPEG2000Codec::GetQuality (unsigned int idx = 0) const`

25.152.3.8 `double gdcmm::JPEG2000Codec::GetRate (unsigned int idx = 0) const`

25.152.3.9 `void gdcmm::JPEG2000Codec::SetNumberOfResolutions (unsigned int nres)`

25.152.3.10 `void gdcmm::JPEG2000Codec::SetQuality (unsigned int idx, double q)`

25.152.3.11 `void gdcmm::JPEG2000Codec::SetRate (unsigned int idx, double rate)`

25.152.3.12 `void gdcmm::JPEG2000Codec::SetReversible (bool res)`

25.152.3.13 `void gdcmm::JPEG2000Codec::SetTileSize (unsigned int tx, unsigned int ty)`

25.152.4 Friends And Related Function Documentation

25.152.4.1 `friend class Bitmap` [friend]

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

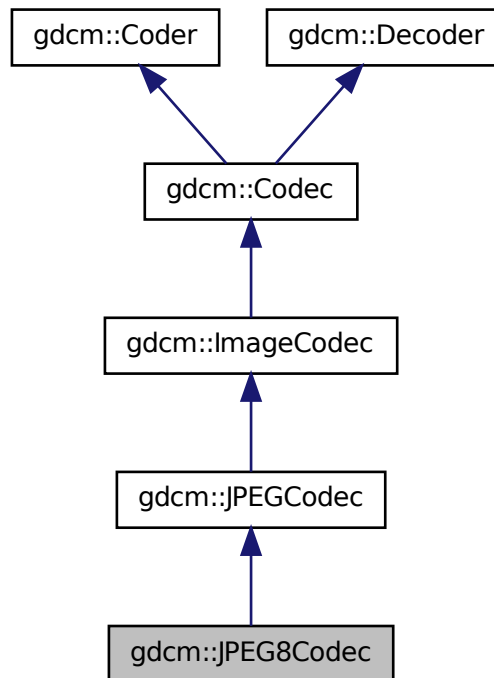
- `gdcmmJPEG2000Codec.h`

25.153 gdcmm::JPEG8Codec Class Reference

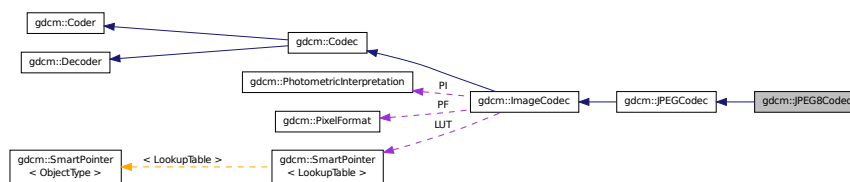
Class to do JPEG 8bits (lossy & lossless)

```
#include <gdcmmJPEG8Codec.h>
```

Inheritance diagram for gdcm::JPEG8Codec:



Collaboration diagram for gdcm::JPEG8Codec:



Public Member Functions

- `JPEG8Codec ()`
- `~JPEG8Codec ()`
- `bool Decode (std::istream &is, std::ostream &os)`
- `bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)`
- `bool InternalCode (const char *input, unsigned long len, std::ostream &os)`

Additional Inherited Members

25.153.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

Note

internal class

25.153.2 Constructor & Destructor Documentation

25.153.2.1 `gdcm::JPEG8Codec::JPEG8Codec ()`

25.153.2.2 `gdcm::JPEG8Codec::~~JPEG8Codec ()`

25.153.3 Member Function Documentation

25.153.3.1 `bool gdcm::JPEG8Codec::Decode (std::istream & is, std::ostream & os)` `[virtual]`

Reimplemented from `gdcm::JPEGCodec`.

25.153.3.2 `bool gdcm::JPEG8Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` `[virtual]`

Reimplemented from `gdcm::JPEGCodec`.

25.153.3.3 `bool gdcm::JPEG8Codec::InternalCode (const char * input, unsigned long len, std::ostream & os)` `[virtual]`

Reimplemented from `gdcm::Coder`.

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

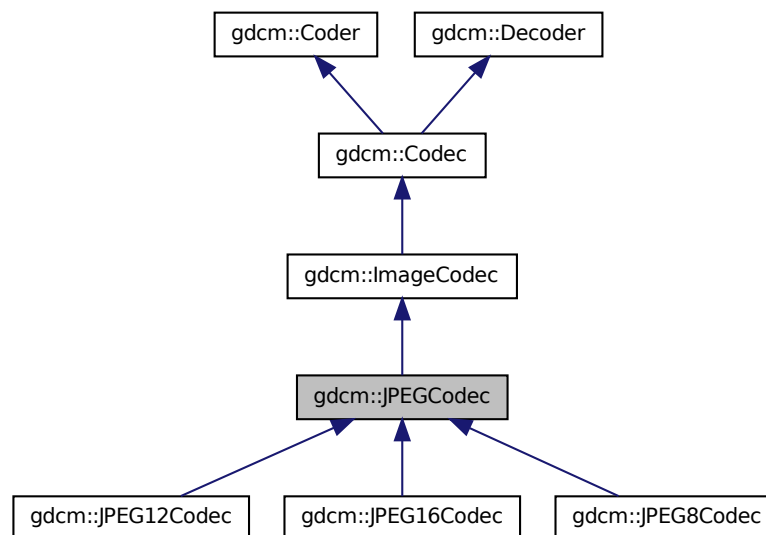
- `gdcmJPEG8Codec.h`

25.154 gdcm::JPEGCodec Class Reference

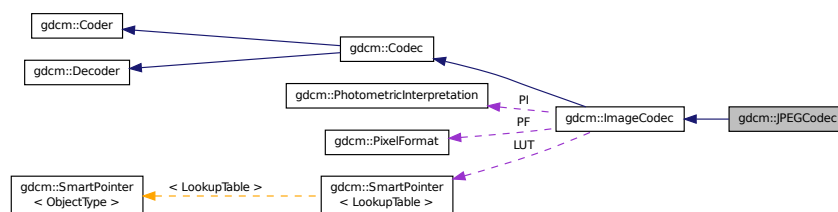
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: `gdcm::JPEG8Codec`, `gdcm::JPEG12Codec` & `gdcm::JPEG16Codec` It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for 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)
- `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 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 Decode (std::istream &is, std::ostream &os)
- bool IsValid (PhotometricInterpretation const &pi)
- void SetBitSample (int bit)

Protected Attributes

- int BitSample
- bool Lossless
- int Quality

Additional Inherited Members

25.154.1 Detailed Description

JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: gdcm::JPEG8Codec, gdcm::JPEG12Codec & gdcm::JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.

Note

Things you should know if you ever want to dive into DICOM/JPEG world (among other):

- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/625e46919f208
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/75fdfccc65a624
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f093
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c92

Examples:

GetJPEGSamplePrecision.cxx.

25.154.2 Constructor & Destructor Documentation

25.154.2.1 gdcm::JPEGCodec::JPEGCodec ()

25.154.2.2 gdcm::JPEGCodec::~~JPEGCodec ()

25.154.3 Member Function Documentation

25.154.3.1 `bool gdcm::JPEGCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from `gdcm::ImageCodec`.

25.154.3.2 `bool gdcm::JPEGCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from `gdcm::ImageCodec`.

25.154.3.3 `bool gdcm::JPEGCodec::Code (DataElement const & in, DataElement & out)` [virtual]

Compress into JPEG.

Reimplemented from `gdcm::Coder`.

25.154.3.4 `void gdcm::JPEGCodec::ComputeOffsetTable (bool b)`

Compute the offset table:

25.154.3.5 `bool gdcm::JPEGCodec::Decode (DataElement const & is_, DataElement & os)` [virtual]

Decode.

Reimplemented from `gdcm::ImageCodec`.

25.154.3.6 `bool gdcm::JPEGCodec::Decode (std::istream & is, std::ostream & os)` [protected],[virtual]

Reimplemented from `gdcm::ImageCodec`.

Reimplemented in `gdcm::JPEG12Codec`, `gdcm::JPEG16Codec`, and `gdcm::JPEG8Codec`.

25.154.3.7 `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.

25.154.3.8 `bool gdcm::JPEGCodec::GetLossless () const`

25.154.3.9 `double gdcm::JPEGCodec::GetQuality () const`

25.154.3.10 `bool gdcM::JPEGCodec::IsValid (PhotometricInterpretation const & pi)` [protected],[virtual]

Reimplemented from `gdcM::ImageCodec`.

25.154.3.11 `void gdcM::JPEGCodec::SetBitSample (int bit)` [protected]

25.154.3.12 `void gdcM::JPEGCodec::SetLossless (bool l)`

25.154.3.13 `void gdcM::JPEGCodec::SetPixelFormat (PixelFormat const & pf)` [virtual]

Reimplemented from `gdcM::ImageCodec`.

Examples:

GetJPEGSamplePrecision.cxx.

25.154.3.14 `void gdcM::JPEGCodec::SetQuality (double q)`

25.154.4 Member Data Documentation

25.154.4.1 `int gdcM::JPEGCodec::BitSample` [protected]

25.154.4.2 `bool gdcM::JPEGCodec::Lossless` [protected]

25.154.4.3 `int gdcM::JPEGCodec::Quality` [protected]

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

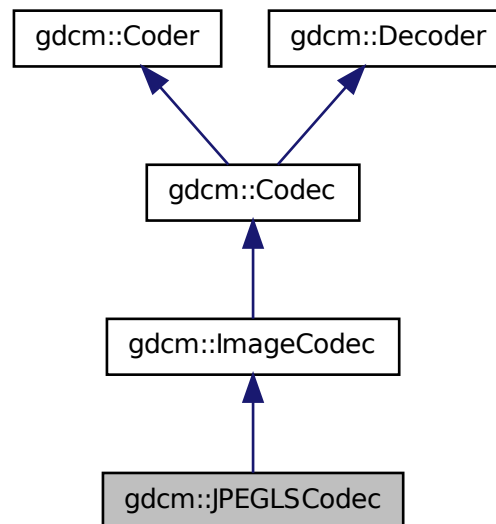
- `gdcMJPEGCodec.h`

25.155 gdcM::JPEGLSCodec Class Reference

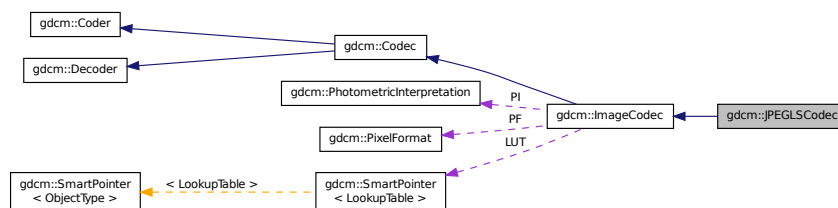
JPEG-LS.

```
#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)
- `bool Code (DataElement const &in, DataElement &out)`
Code.
- `bool Decode (DataElement const &is, DataElement &os)`

Decode.

- `bool Decode (DataElement const &in, char *outBuffer, uint32_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

Additional Inherited Members

25.155.1 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>

25.155.2 Constructor & Destructor Documentation

25.155.2.1 `gdcm::JPEGLSCodec::JPEGLSCodec ()`

25.155.2.2 `gdcm::JPEGLSCodec::~~JPEGLSCodec ()`

25.155.3 Member Function Documentation

25.155.3.1 `bool gdcm::JPEGLSCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from `gdcm::ImageCodec`.

25.155.3.2 `bool gdcm::JPEGLSCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from `gdcm::ImageCodec`.

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

Code.

Reimplemented from `gdcm::Coder`.

25.155.3.4 `bool gdcm::JPEGLSCodec::Decode (DataElement const & is, DataElement & os)` [virtual]

Decode.

Reimplemented from `gdcm::ImageCodec`.

25.155.3.5 `bool gdcm::JPEGLSCodec::Decode (DataElement const & in, char * outBuffer, uint32_t inBufferLength, uint32_t inXMin, uint32_t inXMax, uint32_t inYMin, uint32_t inYMax, uint32_t inZMin, uint32_t inZMax)`

25.155.3.6 `unsigned long gdcm::JPEGLSCodec::GetBufferLength () const` [inline]

25.155.3.7 `bool gdcm::JPEGLSCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from `gdcm::ImageCodec`.

25.155.3.8 `bool gdcm::JPEGLSCodec::GetLossless () const`

25.155.3.9 `void gdcm::JPEGLSCodec::SetBufferLength (unsigned long l)` [inline]

25.155.3.10 `void gdcm::JPEGLSCodec::SetLossless (bool l)`

25.155.3.11 `void gdcm::JPEGLSCodec::SetLossyError (int error)`

[0-3] generally

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

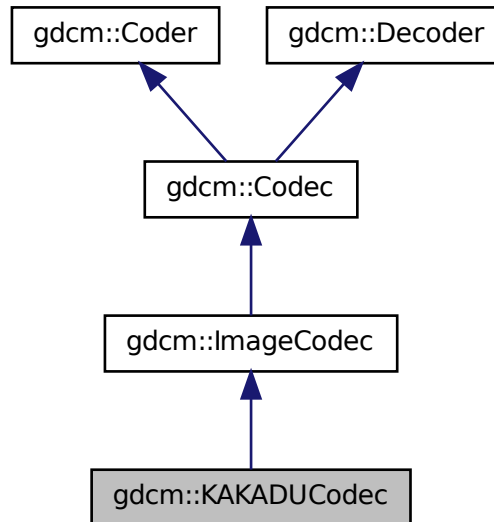
- `gdcmJPEGLSCodec.h`

25.156 gdcm::KAKADUCodec Class Reference

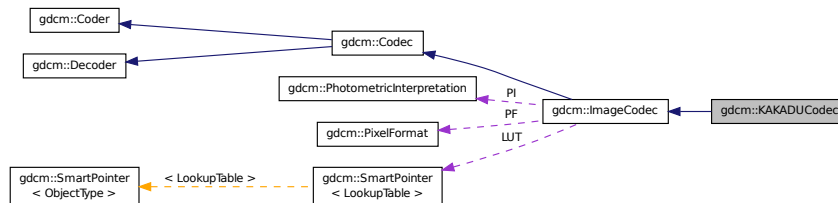
KAKADUCodec.

```
#include <gdcmKAKADUCodec.h>
```

Inheritance diagram for `gdcm::KAKADUCodec`:



Collaboration diagram for `gdcm::KAKADUCodec`:



Public Member Functions

- `KAKADUCodec ()`
- `~KAKADUCodec ()`
- `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)
- `bool Code (DataElement const &in, DataElement &out)`
Code.
- `bool Decode (DataElement const &is, DataElement &os)`
Decode.

Additional Inherited Members

25.156.1 Detailed Description

KAKADUCodec.

25.156.2 Constructor & Destructor Documentation

25.156.2.1 `gdcm::KAKADUCodec::KAKADUCodec ()`

25.156.2.2 `gdcm::KAKADUCodec::~~KAKADUCodec ()`

25.156.3 Member Function Documentation

25.156.3.1 `bool gdcm::KAKADUCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from `gdcm::ImageCodec`.

25.156.3.2 `bool gdcm::KAKADUCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from `gdcm::ImageCodec`.

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

Code.

Reimplemented from `gdcm::Coder`.

25.156.3.4 `bool gdcm::KAKADUCodec::Decode (DataElement const & is_, DataElement & os)` [virtual]

Decode.

Reimplemented from `gdcm::ImageCodec`.

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

- `gdcmKAKADUCodec.h`

25.157 gdcm::LO Class Reference

LO.

```
#include <gdcmLO.h>
```

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=npos)
- bool IsValid () const

25.157.1 Detailed Description

LO.

Note

TODO

25.157.2 Member Typedef Documentation

- 25.157.2.1 typedef Superclass::const_iterator gdcm::LO::const_iterator
- 25.157.2.2 typedef Superclass::const_reference gdcm::LO::const_reference
- 25.157.2.3 typedef Superclass::const_reverse_iterator gdcm::LO::const_reverse_iterator
- 25.157.2.4 typedef Superclass::difference_type gdcm::LO::difference_type
- 25.157.2.5 typedef Superclass::iterator gdcm::LO::iterator
- 25.157.2.6 typedef Superclass::pointer gdcm::LO::pointer
- 25.157.2.7 typedef Superclass::reference gdcm::LO::reference
- 25.157.2.8 typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator
- 25.157.2.9 typedef Superclass::size_type gdcm::LO::size_type
- 25.157.2.10 typedef String<'\',64> gdcm::LO::Superclass

25.157.2.11 `typedef Superclass::value_type gdcm::LO::value_type`

25.157.3 Constructor & Destructor Documentation

25.157.3.1 `gdcm::LO::LO ()` `[inline]`

25.157.3.2 `gdcm::LO::LO (const value_type * s)` `[inline]`

25.157.3.3 `gdcm::LO::LO (const value_type * s, size_type n)` `[inline]`

25.157.3.4 `gdcm::LO::LO (const Superclass & s, size_type pos = 0, size_type n = npos)` `[inline]`

25.157.4 Member Function Documentation

25.157.4.1 `bool gdcm::LO::IsValid () const` `[inline]`

References `gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid()`.

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

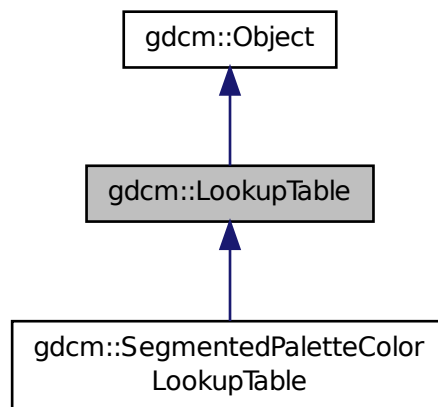
- `gdcmLO.h`

25.158 gdcm::LookupTable Class Reference

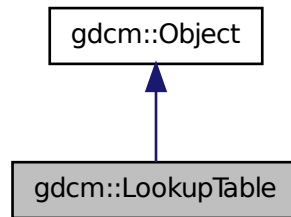
LookupTable class.

```
#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.
- `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

25.158.1 Detailed Description

LookupTable class.

25.158.2 Member Enumeration Documentation

25.158.2.1 enum gdcm::LookupTable::LookupTableType

Enumerator:

RED

GREEN

BLUE

GRAY

UNKNOWN

25.158.3 Constructor & Destructor Documentation

25.158.3.1 gdcm::LookupTable::LookupTable ()

25.158.3.2 gdcm::LookupTable::~~LookupTable ()

25.158.3.3 gdcm::LookupTable::LookupTable (LookupTable const & lut) [inline]

25.158.4 Member Function Documentation

25.158.4.1 void gdcmm::LookupTable::Allocate (unsigned short *bitsample* = 8)

Allocate the LUT.

25.158.4.2 void gdcmm::LookupTable::Clear ()

Clear the LUT.

25.158.4.3 void gdcmm::LookupTable::Decode (std::istream & *is*, std::ostream & *os*) const

Decode the LUT.

25.158.4.4 unsigned short gdcmm::LookupTable::GetBitSample () const [inline]

return the bit sample

25.158.4.5 bool gdcmm::LookupTable::GetBufferAsRGBA (unsigned char * *rgba*) const

return the LUT as RGBA buffer

25.158.4.6 void gdcmm::LookupTable::GetLUT (LookupTableType *type*, unsigned char * *array*, unsigned int & *length*) const

25.158.4.7 void gdcmm::LookupTable::GetLUTDescriptor (LookupTableType *type*, unsigned short & *length*, unsigned short & *subscript*, unsigned short & *bitsize*) const

25.158.4.8 unsigned int gdcmm::LookupTable::GetLUTLength (LookupTableType *type*) const

25.158.4.9 const unsigned char* gdcmm::LookupTable::GetPointer () const

return a raw pointer to the LUT

25.158.4.10 void gdcmm::LookupTable::InitializeBlueLUT (unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize*)

25.158.4.11 bool gdcmm::LookupTable::Initialized () const

return whether the LUT has been initialized

25.158.4.12 void gdcmm::LookupTable::InitializeGreenLUT (unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize*)

25.158.4.13 void gdcmm::LookupTable::InitializeLUT (LookupTableType *type*, unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize*)

Generic interface:

25.158.4.14 void gdcmm::LookupTable::InitializeRedLUT (unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize*)

RED / GREEN / BLUE specific:

25.158.4.15 `void gdcm::LookupTable::Print (std::ostream &) const` `[inline]`,`[virtual]`

Reimplemented from `gdcm::Object`.

Reimplemented in `gdcm::SegmentedPaletteColorLookupTable`.

25.158.4.16 `void gdcm::LookupTable::SetBlueLUT (const unsigned char * blue, unsigned int length)`

25.158.4.17 `void gdcm::LookupTable::SetGreenLUT (const unsigned char * green, unsigned int length)`

25.158.4.18 `virtual void gdcm::LookupTable::SetLUT (LookupTableType type, const unsigned char * array, unsigned int length)` `[virtual]`

Reimplemented in `gdcm::SegmentedPaletteColorLookupTable`.

25.158.4.19 `void gdcm::LookupTable::SetRedLUT (const unsigned char * red, unsigned int length)`

25.158.4.20 `bool gdcm::LookupTable::WriteBufferAsRGBA (const unsigned char * rgba)`

Write the LUT as RGBA.

25.158.5 Member Data Documentation

25.158.5.1 `unsigned short gdcm::LookupTable::BitSample` `[protected]`

25.158.5.2 `bool gdcm::LookupTable::IncompleteLUT` `[protected]`

25.158.5.3 `LookupTableInternal* gdcm::LookupTable::Internal` `[protected]`

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

- `gdcmLookupTable.h`

25.159 gdcm::Scanner::Itstr Struct Reference

```
#include <gdcmScanner.h>
```

Public Member Functions

- `bool operator() (const char *s1, const char *s2) const`

25.159.1 Member Function Documentation

25.159.1.1 `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`

25.160 gdcmmacro Class Reference

Class for representing a Macro.

```
#include <gdcmmacro.h>
```

Public Types

- typedef std::vector< std::string > ArrayIncludeMacrosType
- typedef std::map< Tag, MacroEntry > MapModuleEntry

Public Member Functions

- Macro ()
- 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)

25.160.1 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

25.160.2 Member Typedef Documentation

25.160.2.1 typedef std::vector<std::string> gdcmmacro::ArrayIncludeMacrosType

25.160.2.2 typedef std::map<Tag, MacroEntry> gdcmmacro::MapModuleEntry

25.160.3 Constructor & Destructor Documentation

25.160.3.1 gdcmmacro::Macro () [inline]

25.160.4 Member Function Documentation

25.160.4.1 void gdcmmacros::Macro::AddMacroEntry (const Tag & tag, const MacroEntry & module) [inline]

Will add a ModuleEntry directly at root-level. See Macro for nested-included level.

25.160.4.2 void gdcmmacros::Macro::Clear () [inline]

25.160.4.3 bool gdcmmacros::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.

25.160.4.4 const MacroEntry& gdcmmacros::Macro::GetMacroEntry (const Tag & tag) const

25.160.4.5 const char* gdcmmacros::Macro::GetName () const [inline]

25.160.4.6 void gdcmmacros::Macro::SetName (const char * name) [inline]

25.160.4.7 bool gdcmmacros::Macro::Verify (const DataSet & ds, Usage const & usage) const

25.160.5 Friends And Related Function Documentation

25.160.5.1 std::ostream& operator<< (std::ostream & os, const Macro & val) [friend]

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

- gdcmmacros.h

25.161 gdcmmacros 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)`

25.161.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples:

TraverseModules.cxx.

25.161.2 Member Typedef Documentation

25.161.2.1 `typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType`

25.161.3 Constructor & Destructor Documentation

25.161.3.1 `gdcm::Macros::Macros ()` `[inline]`

25.161.4 Member Function Documentation

25.161.4.1 `void gdcm::Macros::AddMacro (const char * ref, const Macro & module)` `[inline]`

25.161.4.2 `void gdcm::Macros::Clear ()` `[inline]`

25.161.4.3 `const Macro& gdcm::Macros::GetMacro (const char * name) const` `[inline]`

25.161.4.4 `bool gdcm::Macros::IsEmpty () const` `[inline]`

25.161.5 Friends And Related Function Documentation

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

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

- `gdcmMacros.h`

25.162 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`
- `std::istream & Read (std::istream &is)`
- `void SetMaximumLength (uint32_t maximumlength)`
- `size_t Size () const`
- `const std::ostream & Write (std::ostream &os) const`

25.162.1 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)

25.162.2 Constructor & Destructor Documentation

25.162.2.1 `gdcm::network::MaximumLengthSub::MaximumLengthSub ()`

25.162.3 Member Function Documentation

25.162.3.1 `uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength () const` `[inline]`

25.162.3.2 `std::istream& gdcm::network::MaximumLengthSub::Read (std::istream & is)`

25.162.3.3 `void gdcm::network::MaximumLengthSub::SetMaximumLength (uint32_t maximumlength)` `[inline]`

25.162.3.4 `size_t gdcm::network::MaximumLengthSub::Size () const`

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

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

- `gdcmMaximumLengthSub.h`

25.163 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])

25.163.1 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

25.163.2 Constructor & Destructor Documentation

25.163.2.1 gdcM::MD5::MD5 ()

25.163.2.2 gdcM::MD5::~~MD5 ()

25.163.3 Member Function Documentation

25.163.3.1 static bool gdcM::MD5::Compute (const char * *buffer*, unsigned long *buf_len*, char *digest_str*[33]) [static]

25.163.3.2 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

25.164 gdcM::MediaStorage Class Reference

MediaStorage.

```
#include <gdcMMediaStorage.h>
```

Public Types

- enum MType {
 - 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,
 - EnhancedDICOMStorage,
 - StudyComponentManagementSOPClass,
 - DetachedVisitManagementSOPClass,
 - DetachedPatientManagementSOPClass,
 - VideoEndoscopicImageStorage,

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)

25.164.1 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, Stream-ImageReaderTest.cxx, and TestReader.cxx.

25.164.2 Member Enumeration Documentation

25.164.2.1 enum gdcmm::MediaStorage::MSType

Enumerator:

MediaStorageDirectoryStorage
ComputedRadiographylmageStorage
DigitalXRayImageStorageForPresentation
DigitalXRayImageStorageForProcessing
DigitalMammographylmageStorageForPresentation
DigitalMammographylmageStorageForProcessing
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

StudyComponentManagementSOPClass

DetachedVisitManagementSOPClass

DetachedPatientManagementSOPClass

VideoEndoscopicImageStorage

GeneralElectricMagneticResonanceImageStorage

GEPrivate3DModelStorage

ToshibaPrivateDataStorage

MammographyCADSR

KeyObjectSelectionDocument

HangingProtocolStorage
ModalityPerformedProcedureStepSOPClass
PhilipsPrivateMRSyntheticImageStorage
VLPhotographicImageStorage
SegmentationStorage
RTIonPlanStorage
XR3DAniographicImageStorage
EnhancedXAImageStorage
RTIonBeamsTreatmentRecordStorage
SurfaceSegmentationStorage
VLWholeSlideMicroscopyImageStorage
MS_END

Examples:

GenerateStandardSOPClasses.cxx.

25.164.2.2 enum gdcm::MediaStorage::ObjectType

Enumerator:

NoObject
Video
Waveform
Audio
PDF
URI
Segmentation
ObjectEnd

25.164.3 Constructor & Destructor Documentation

25.164.3.1 gdcm::MediaStorage::MediaStorage (MStype type = MS_END) [inline]

25.164.4 Member Function Documentation

25.164.4.1 const char* gdcm::MediaStorage::GetModality () const

25.164.4.2 unsigned int gdcm::MediaStorage::GetModalityDimension () const

25.164.4.3 static const char* gdcm::MediaStorage::GetMSString (MStype ts) [static]

Return the Media String associated. Will return NULL for MS_END.

Examples:

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

25.164.4.4 `static MType gdcM::MediaStorage::GetMType (const char * str) [static]`

Examples:

TestReader.cxx.

25.164.4.5 `static unsigned int gdcM::MediaStorage::GetNumberOfModality () [static]`

25.164.4.6 `static unsigned int gdcM::MediaStorage::GetNumberOfMSString () [static]`

25.164.4.7 `static unsigned int gdcM::MediaStorage::GetNumberOfMType () [static]`

25.164.4.8 `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, GetSub-SequenceData.cxx, iU22tomultisc.cxx, and StreamImageReaderTest.cxx.

25.164.4.9 `void gdcM::MediaStorage::GuessFromModality (const char * modality, unsigned int dimension = 2)`

25.164.4.10 `static bool gdcM::MediaStorage::IsImage (MType ts) [static]`

Returns whether DICOM has a Pixel Data element (7fe0,0010)

Warning

MRSpectroscopyStorage could be image but are not

25.164.4.11 `bool gdcM::MediaStorage::IsUndefined () const [inline]`

Examples:

TestReader.cxx.

25.164.4.12 `gdcM::MediaStorage::operator MType () const [inline]`

25.164.4.13 `bool gdcM::MediaStorage::SetFromDataSet (DataSet const & ds)`

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

25.164.4.14 `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.

25.164.4.15 `bool gdcm::MediaStorage::SetFromHeader (FileMetaInformation const & fmi)`

25.164.4.16 `bool gdcm::MediaStorage::SetFromModality (DataSet const & ds)`

25.164.4.17 `void gdcm::MediaStorage::SetFromSourceImageSequence (DataSet const & ds)` [protected]

25.164.5 Friends And Related Function Documentation

25.164.5.1 `std::ostream& operator<< (std::ostream & os, const MediaStorage & ms)` [friend]

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

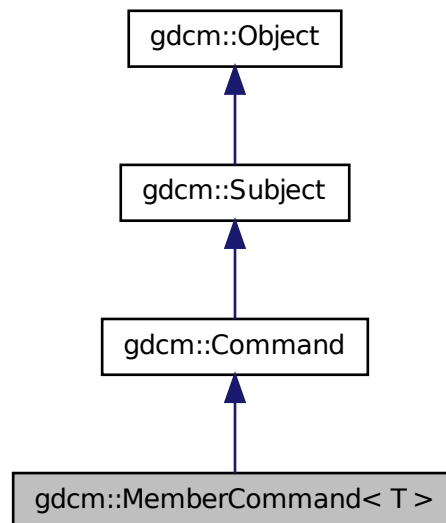
- gdcmMediaStorage.h

25.165 gdcm::MemberCommand< T > Class Template Reference

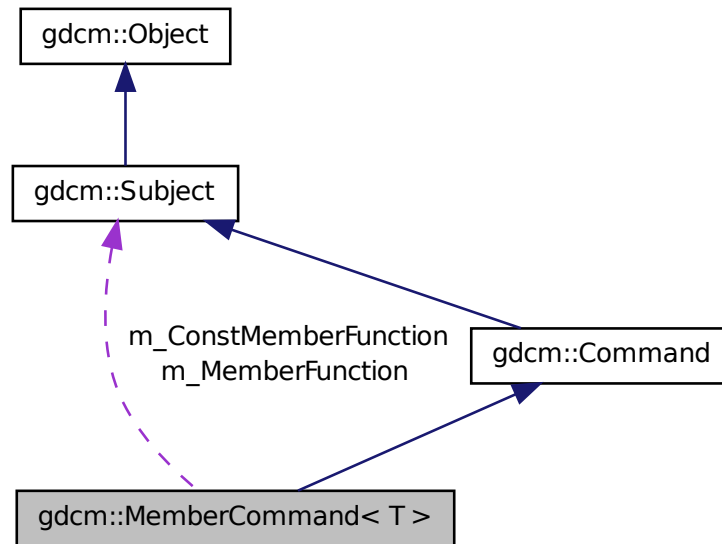
Command subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::MemberCommand< T >:



Collaboration diagram for `gdcm::MemberCommand< T >`:



Public Types

- `typedef MemberCommand Self`
- `typedef void(T::* TConstMemberFunctionPointer)(const Subject *, const Event &)`
- `typedef void(T::* TMemberFunctionPointer)(Subject *, const Event &)`

Public Member Functions

- `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

25.165.1 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.

25.165.2 Member Typedef Documentation

25.165.2.1 template<class T > typedef MemberCommand gdcmmembercommand< T >::Self

Standard class typedefs.

25.165.2.2 template<class T > typedef void(T::* gdcmmembercommand< T >::TConstMemberFunctionPointer)(const Subject *, const Event &)

25.165.2.3 template<class T > typedef void(T::* gdcmmembercommand< T >::TMemberFunctionPointer)(Subject *, const Event &)

pointer to a member function that takes a Subject* and the event

25.165.3 Constructor & Destructor Documentation

25.165.3.1 template<class T > gdcmmembercommand< T >::MemberCommand () [inline],
[protected]

Referenced by gdcmmembercommand< T >::New().

25.165.3.2 template<class T > virtual gdcmmembercommand< T >::~~MemberCommand () [inline],
[protected], [virtual]

25.165.4 Member Function Documentation

25.165.4.1 template<class T > virtual void gdcmmembercommand< T >::Execute (Subject * caller, const Event & event) [inline], [virtual]

Invoke the member function.

Implements gdcmmembercommand.

References gdcmmembercommand< T >::m_MemberFunction.

25.165.4.2 `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.

25.165.4.3 `template<class T> static SmartPointer<MemberCommand> gdcM::MemberCommand< T>::New () [inline],[static]`

Method for creation through the object factory.

References gdcM::MemberCommand< T>::MemberCommand().

25.165.4.4 `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.

25.165.4.5 `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.

25.165.5 Member Data Documentation

25.165.5.1 `template<class T> TConstMemberFunctionPointer gdcM::MemberCommand< T>::m_ConstMemberFunction [protected]`

Referenced by gdcM::MemberCommand< T>::Execute(), and gdcM::MemberCommand< T>::SetCallbackFunction().

25.165.5.2 `template<class T> TMemberFunctionPointer gdcM::MemberCommand< T>::m_MemberFunction [protected]`

Referenced by gdcM::MemberCommand< T>::Execute(), and gdcM::MemberCommand< T>::SetCallbackFunction().

25.165.5.3 `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:

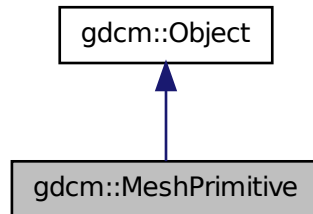
- gdcMCommand.h

25.166 gdcmmesh::MeshPrimitive Class Reference

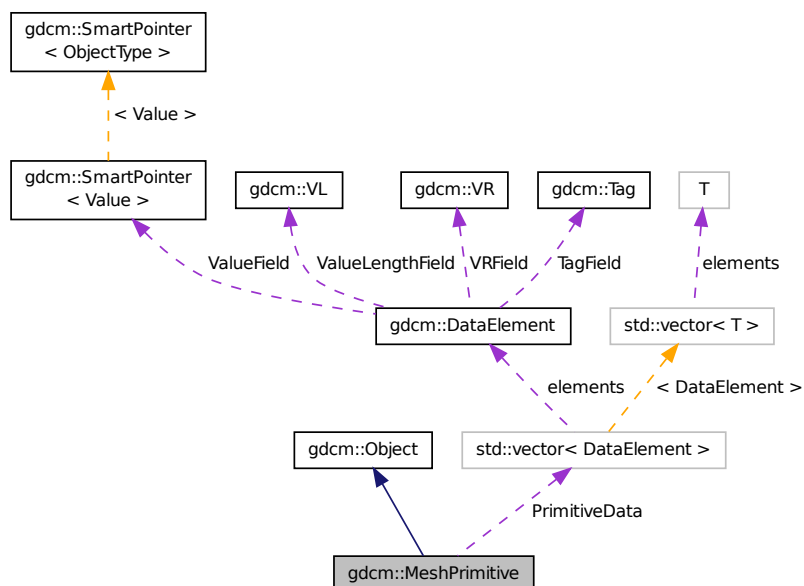
This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

```
#include <gdcmMeshPrimitive.h>
```

Inheritance diagram for `gdc::MeshPrimitive`:



Collaboration diagram for `gdc::MeshPrimitive`:



Public Types

- enum MPType {
 VERTEX = 0,
 EDGE,
 TRIANGLE,
 TRIANGLE_STRIP,
 TRIANGLE_FAN,
 LINE,
 FACET,
 MPType_END }
- *This enumeration defines primitive types.*
- typedef std::vector< DataElement > PrimitivesData

Public Member Functions

- MeshPrimitive ()
- 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 ()
- MPType GetPrimitiveType () const
- void SetPrimitiveData (DataElement const &de)
- void SetPrimitiveData (const unsigned int idx, DataElement const &de)
- void SetPrimitivesData (PrimitivesData const &DEs)
- void SetPrimitiveType (const MPType type)

Static Public Member Functions

- static MPType GetMPType (const char *type)
- static const char * GetMPTypeString (const MPType type)

Protected Attributes

- PrimitivesData PrimitiveData
- MPType PrimitiveType

Additional Inherited Members

25.166.1 Detailed Description

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

25.166.2 Member Typedef Documentation

25.166.2.1 `typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData`

25.166.3 Member Enumeration Documentation

25.166.3.1 `enum gdcm::MeshPrimitive::MPType`

This enumeration defines primitive types.

See also

PS 3.3 C.27.4.1

Enumerator:

VERTEX
EDGE
TRIANGLE
TRIANGLE_STRIP
TRIANGLE_FAN
LINE
FACET
MPType_END

25.166.4 Constructor & Destructor Documentation

25.166.4.1 `gdcm::MeshPrimitive::MeshPrimitive ()`

25.166.4.2 `virtual gdcm::MeshPrimitive::~~MeshPrimitive ()` [virtual]

25.166.5 Member Function Documentation

25.166.5.1 `void gdcm::MeshPrimitive::AddPrimitiveData (DataElement const & de)`

25.166.5.2 `static MPType gdcm::MeshPrimitive::GetMPType (const char * type)` [static]

25.166.5.3 `static const char* gdcm::MeshPrimitive::GetMPTypeString (const MPType type)` [static]

25.166.5.4 `unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ()` const

25.166.5.5 `const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ()` const

25.166.5.6 `DataElement& gdcm::MeshPrimitive::GetPrimitiveData ()`

25.166.5.7 `const DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int idx)` const

25.166.5.8 `DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int idx)`

25.166.5.9 `const PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ()` const

25.166.5.10 **PrimitivesData&** **gdcm::MeshPrimitive::GetPrimitivesData** ()

25.166.5.11 **MPType** **gdcm::MeshPrimitive::GetPrimitiveType** () const

25.166.5.12 void **gdcm::MeshPrimitive::SetPrimitiveData** (**DataElement** const & *de*)

25.166.5.13 void **gdcm::MeshPrimitive::SetPrimitiveData** (const unsigned int *idx*, **DataElement** const & *de*)

25.166.5.14 void **gdcm::MeshPrimitive::SetPrimitivesData** (**PrimitivesData** const & *DEs*)

25.166.5.15 void **gdcm::MeshPrimitive::SetPrimitiveType** (const **MPType** *type*)

25.166.6 Member Data Documentation

25.166.6.1 **PrimitivesData** **gdcm::MeshPrimitive::PrimitiveData** [protected]

25.166.6.2 **MPType** **gdcm::MeshPrimitive::PrimitiveType** [protected]

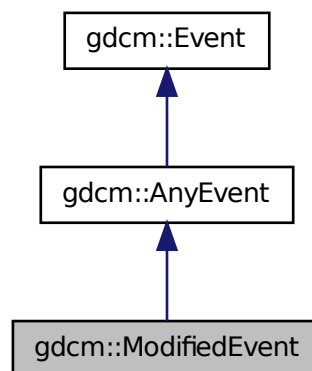
The documentation for this class was generated from the following file:

- `gdcmMeshPrimitive.h`

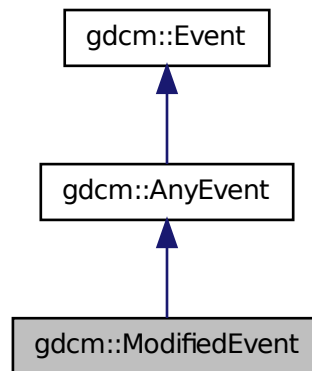
25.167 gdcm::ModifiedEvent Class Reference

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

Inheritance diagram for `gdcm::ModifiedEvent`:



Collaboration diagram for gdcm::ModifiedEvent:



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

- `gdcmEvent.h`

25.168 gdcm::Module Class Reference

Class for representing a Module.

```
#include <gdcmModule.h>
```

Public Types

- `typedef std::vector< std::string > ArrayIncludeMacrosType`
- `typedef std::map< Tag, ModuleEntry > MapModuleEntry`

Public Member Functions

- `Module ()`
- `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)`

25.168.1 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.

25.168.2 Member Typedef Documentation

25.168.2.1 `typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacrosType`

25.168.2.2 `typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry`

25.168.3 Constructor & Destructor Documentation

25.168.3.1 `gdcm::Module::Module () [inline]`

25.168.4 Member Function Documentation

25.168.4.1 `void gdcm::Module::AddMacro (const char * include) [inline]`

25.168.4.2 `void gdcm::Module::AddModuleEntry (const Tag & tag, const ModuleEntry & module) [inline]`

Will add a ModuleEntry directly at root-level. See Macro for nested-included level.

25.168.4.3 `void gdcm::Module::Clear () [inline]`

25.168.4.4 `bool gdcm::Module::FindModuleEntryInMacros (Macros const & macros, const Tag & tag) const`

Find or Get a ModuleEntry. ModuleEntry are either search are root-level or within nested-macro included in module.

Examples:

TraverseModules.cxx.

25.168.4.5 `const ModuleEntry& gdcm::Module::GetModuleEntryInMacros (Macros const & macros, const Tag & tag) const`

Examples:

TraverseModules.cxx.

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

25.168.4.7 `void gdcm::Module::SetName (const char * name)` `[inline]`

25.168.4.8 `bool gdcm::Module::Verify (const DataSet & ds, Usage const & usage) const`

25.168.5 Friends And Related Function Documentation

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

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

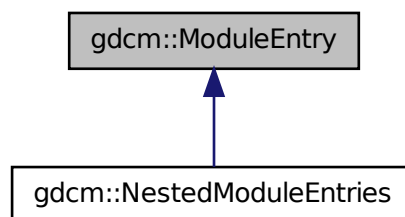
- `gdcmModule.h`

25.169 gdcm::ModuleEntry Class Reference

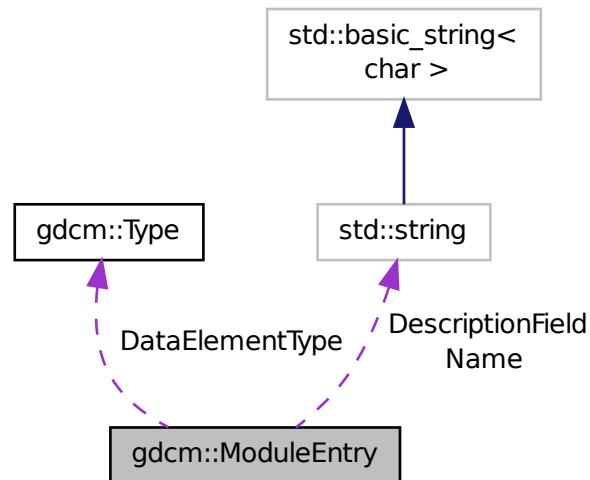
Class for representing a ModuleEntry.

```
#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)`

25.169.1 Detailed Description

Class for representing a ModuleEntry.

Note

bla

See also

DictEntry

Examples:

TraverseModules.cxx.

25.169.2 Member Typedef Documentation

25.169.2.1 `typedef std::string gdcmmoduleentry::Description`

25.169.3 Constructor & Destructor Documentation

25.169.3.1 `gdcmmoduleentry::ModuleEntry (const char * name = " ", const char * type = "3", const char * description = " ")`
`[inline]`

References `gdcmmoduleentry::Type::GetTypeType()`.

25.169.3.2 `virtual gdcmmoduleentry::~ModuleEntry ()` `[inline]`, `[virtual]`

25.169.4 Member Function Documentation

25.169.4.1 `const Description& gdcmmoduleentry::GetDescription () const` `[inline]`

25.169.4.2 `const char* gdcmmoduleentry::GetName () const` `[inline]`

25.169.4.3 `const Type& gdcmmoduleentry::GetType () const` `[inline]`

Examples:

TraverseModules.cxx.

25.169.4.4 `void gdcmmoduleentry::SetDescription (const char * d)` `[inline]`

25.169.4.5 `void gdcmmoduleentry::SetName (const char * name)` `[inline]`

25.169.4.6 `void gdcmmoduleentry::SetType (const Type & type)` `[inline]`

25.169.5 Friends And Related Function Documentation

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

25.169.6 Member Data Documentation

25.169.6.1 Type `gdcm::ModuleEntry::DataElementType` [protected]

Referenced by `gdcm::operator<<()`.

25.169.6.2 Description `gdcm::ModuleEntry::DescriptionField` [protected]

Referenced by `gdcm::operator<<()`.

25.169.6.3 `std::string gdcm::ModuleEntry::Name` [protected]

Referenced by `gdcm::operator<<()`.

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

- `gdcmModuleEntry.h`

25.170 `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)`

25.170.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples:

TraverseModules.cxx.

25.170.2 Member Typedef Documentation

25.170.2.1 `typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType`

25.170.3 Constructor & Destructor Documentation

25.170.3.1 `gdcm::Modules::Modules ()` `[inline]`

25.170.4 Member Function Documentation

25.170.4.1 `void gdcm::Modules::AddModule (const char * ref, const Module & module)` `[inline]`

25.170.4.2 `void gdcm::Modules::Clear ()` `[inline]`

25.170.4.3 `const Module& gdcm::Modules::GetModule (const char * name) const` `[inline]`

25.170.4.4 `bool gdcm::Modules::IsEmpty () const` `[inline]`

25.170.5 Friends And Related Function Documentation

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

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

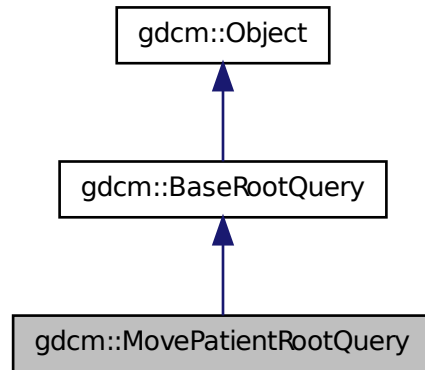
- `gdcmModules.h`

25.171 gdcm::MovePatientRootQuery Class Reference

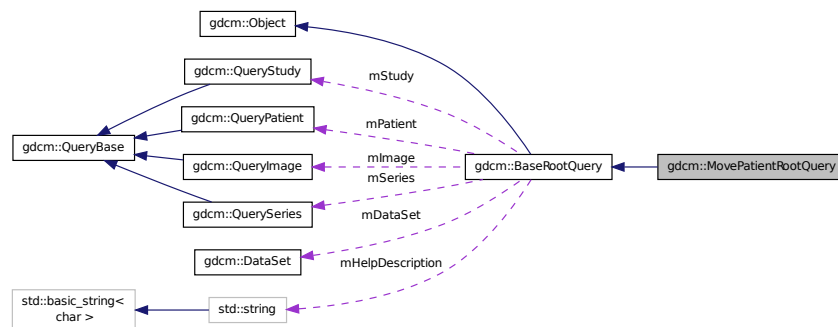
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

```
#include <gdcmMovePatientRootQuery.h>
```

Inheritance diagram for `gdcmm::MovePatientRootQuery`:



Collaboration diagram for `gdcmm::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

25.171.1 Detailed Description

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

25.171.2 Constructor & Destructor Documentation

25.171.2.1 `gdcm::MovePatientRootQuery::MovePatientRootQuery ()`

25.171.3 Member Function Documentation

25.171.3.1 `UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID () const` `[virtual]`

Implements `gdcm::BaseRootQuery`.

25.171.3.2 `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`.

25.171.3.3 `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 dcm4chee

Implements `gdcm::BaseRootQuery`.

25.171.3.4 `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`.

25.171.4 Friends And Related Function Documentation

25.171.4.1 `friend class QueryFactory` `[friend]`

Reimplemented from `gdcm::BaseRootQuery`.

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

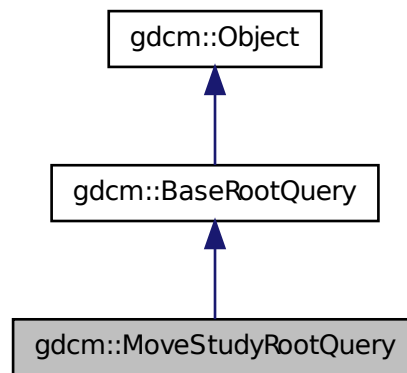
- `gdcmMovePatientRootQuery.h`

25.172 `gdcm::MoveStudyRootQuery` Class Reference

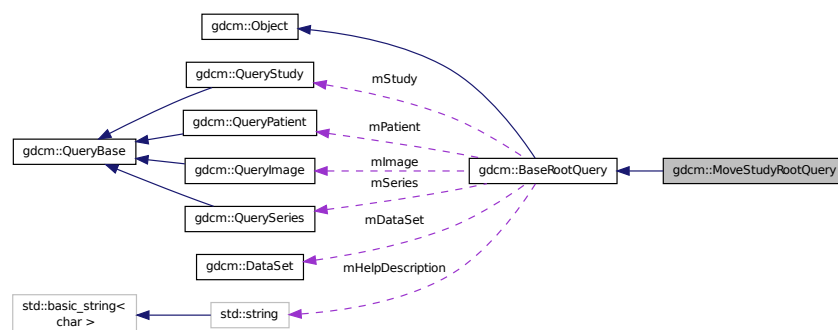
`MoveStudyRootQuery` contains: the class which will produce a dataset for C-MOVE with study root.

```
#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

25.172.1 Detailed Description

MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

25.172.2 Constructor & Destructor Documentation

25.172.2.1 `gdcm::MoveStudyRootQuery::MoveStudyRootQuery ()`

25.172.3 Member Function Documentation

25.172.3.1 `UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID () const` [virtual]

Implements `gdcm::BaseRootQuery`.

25.172.3.2 `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`.

25.172.3.3 `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 dcm4k

Implements `gdcm::BaseRootQuery`.

25.172.3.4 `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`.

25.172.4 Friends And Related Function Documentation

25.172.4.1 friend class QueryFactory [friend]

Reimplemented from gdcM::BaseRootQuery.

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

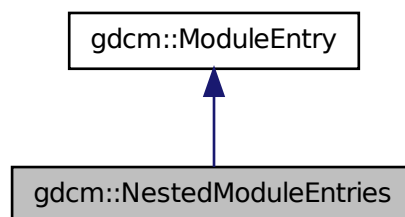
- gdcMMoveStudyRootQuery.h

25.173 gdcM::NestedModuleEntries Class Reference

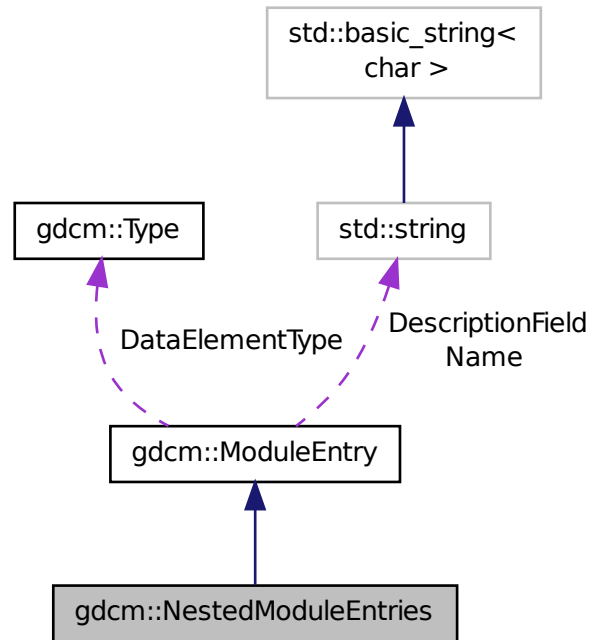
Class for representing a NestedModuleEntries.

```
#include <gdcMNestedModuleEntries.h>
```

Inheritance diagram for gdcM::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



Public Types

- typedef std::vector
 < ModuleEntry >::size_type SizeType

Public Member Functions

- NestedModuleEntries (const char *name="", const char *type="3", const char *description="")
- void AddModuleEntry (const ModuleEntry &me)
- 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

25.173.1 Detailed Description

Class for representing a NestedModuleEntries.

Note

bla

See also

ModuleEntry

25.173.2 Member Typedef Documentation

25.173.2.1 `typedef std::vector<ModuleEntry>::size_type gdcM::NestedModuleEntries::SizeType`

25.173.3 Constructor & Destructor Documentation

25.173.3.1 `gdcM::NestedModuleEntries::NestedModuleEntries (const char * name = " ", const char * type = "3", const char * description = " ") [inline]`

25.173.4 Member Function Documentation

25.173.4.1 `void gdcM::NestedModuleEntries::AddModuleEntry (const ModuleEntry & me) [inline]`

25.173.4.2 `const ModuleEntry& gdcM::NestedModuleEntries::GetModuleEntry (SizeType idx) const [inline]`

25.173.4.3 `ModuleEntry& gdcM::NestedModuleEntries::GetModuleEntry (SizeType idx) [inline]`

25.173.4.4 `SizeType gdcM::NestedModuleEntries::GetNumberOfModuleEntries () [inline]`

25.173.5 Friends And Related Function Documentation

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

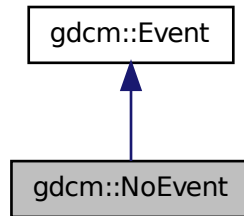
The documentation for this class was generated from the following file:

- `gdcMNestedModuleEntries.h`

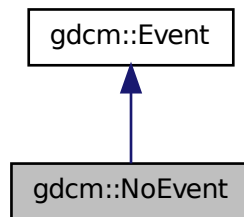
25.174 gdcM::NoEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



Additional Inherited Members

25.174.1 Detailed Description

Define some common GDCM events

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

- gdcmEvent.h

25.175 gdcm::Object Class Reference

Object.

```
#include <gdcmObject.h>
```


25.175.1 Detailed Description

Object.

Note

main superclass for object that want to use SmartPointer invasive ref counting system

See also

SmartPointer

25.175.2 Constructor & Destructor Documentation

25.175.2.1 `gdcm::Object::Object ()` `[inline]`

25.175.2.2 `virtual gdcm::Object::~~Object ()` `[inline]`, `[virtual]`

25.175.2.3 `gdcm::Object::Object (const Object &)` `[inline]`

Special requirement for copy/cstor, assignment operator.

25.175.3 Member Function Documentation

25.175.3.1 `void gdcm::Object::operator= (const Object &)` `[inline]`

25.175.3.2 `virtual void gdcm::Object::Print (std::ostream &) const` `[inline]`, `[virtual]`

Reimplemented in `gdcm::ByteValue`, `gdcm::SequenceOfItems`, `gdcm::SequenceOfFragments`, `gdcm::Scanner`, `gdcm::Image`, `gdcm::Overlay`, `gdcm::Bitmap`, `gdcm::Curve`, `gdcm::LookupTable`, `gdcm::Pixmap`, and `gdcm::SegmentedPaletteColorLookupTable`.

Examples:

`ReadAndDumpDICOMDIR.cxx`.

Referenced by `gdcm::operator<<()`.

25.175.3.3 `void gdcm::Object::Register ()` `[inline]`, `[protected]`

25.175.3.4 `void gdcm::Object::UnRegister ()` `[inline]`, `[protected]`

25.175.4 Friends And Related Function Documentation

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

25.175.4.2 `friend class SmartPointer` `[friend]`

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

- `gdcmObject.h`

25.176 gdcm::OneShotReadBuf Struct Reference

```
#include <gdcmStreamImageReader.h>
```

Public Member Functions

- OneShotReadBuf (void *s, std::size_t n)

25.176.1 Constructor & Destructor Documentation

25.176.1.1 gdcm::OneShotReadBuf::OneShotReadBuf (void * s, std::size_t n) [inline]

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

- gdcmStreamImageReader.h

25.177 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)

25.177.1 Detailed Description

class to handle Orientation

25.177.2 Member Enumeration Documentation

25.177.2.1 enum gdcm::Orientation::OrientationType

Enumerator:

UNKNOWN

AXIAL

CORONAL

SAGITTAL

OBLIQUE

25.177.3 Constructor & Destructor Documentation

25.177.3.1 gdcm::Orientation::Orientation ()

25.177.3.2 gdcm::Orientation::~~Orientation ()

25.177.4 Member Function Documentation

25.177.4.1 static const char* gdcm::Orientation::GetLabel (OrientationType type) [static]

Return the label of an Orientation.

25.177.4.2 static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (double x, double y, double z) [static], [protected]

25.177.4.3 static double gdcm::Orientation::GetObliquityThresholdCosineValue () [static]

25.177.4.4 static OrientationType gdcm::Orientation::GetType (const double dircos[6]) [static]

Return the type of orientation from a direction cosines Input is an array of 6 double

25.177.4.5 `void gdcM::Orientation::Print (std::ostream &) const`

Print.

Referenced by `gdcM::operator<<()`.

25.177.4.6 `static void gdcM::Orientation::SetObliquityThresholdCosineValue (double val) [static]`

ObliquityThresholdCosineValue stuff.

25.177.5 Friends And Related Function Documentation

25.177.5.1 `std::ostream& operator<< (std::ostream & _os, const Orientation & o) [friend]`

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

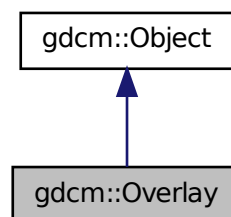
- `gdcMOrientation.h`

25.178 gdcM::Overlay Class Reference

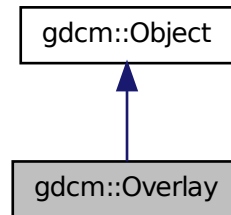
Overlay class.

```
#include <gdcMOverlay.h>
```

Inheritance diagram for `gdcM::Overlay`:



Collaboration diagram for gdcm::Overlay:



Public Member Functions

- Overlay ()
- Overlay (Overlay const &ov)
- ~Overlay ()
- void Decode (std::istream &is, std::ostream &os)
- void Decompress (std::ostream &os) const
- unsigned short GetBitPosition () const
return bit position
- unsigned short GetBitsAllocated () const
return bits allocated
- bool GetBuffer (char *buffer) const
- 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
- bool GetUnpackBuffer (unsigned char *buffer) const
- bool GrabOverlayFromPixelData (DataSet const &ds)
- bool IsEmpty () const
- bool IsInPixelData () const
- void IsInPixelData (bool b)
- bool IsZero () const
return true if all bits are set to 0

- 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)
set origin
- void SetOverlay (const char *array, unsigned int 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:

Additional Inherited Members

25.178.1 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:

25.178.2 Constructor & Destructor Documentation

25.178.2.1 `gdcm::Overlay::Overlay ()`

25.178.2.2 `gdcm::Overlay::~~Overlay ()`

25.178.2.3 `gdcm::Overlay::Overlay (Overlay const & ov)`

25.178.3 Member Function Documentation

25.178.3.1 `void gdcm::Overlay::Decode (std::istream & is, std::ostream & os)`

25.178.3.2 `void gdcm::Overlay::Decompress (std::ostream & os) const`

25.178.3.3 `unsigned short gdcm::Overlay::GetBitPosition () const`

return bit position

25.178.3.4 `unsigned short gdcm::Overlay::GetBitsAllocated () const`

return bits allocated

25.178.3.5 `bool gdcm::Overlay::GetBuffer (char * buffer) const`

25.178.3.6 `unsigned short gdcm::Overlay::GetColumns () const`

get columns

25.178.3.7 `const char* gdcm::Overlay::GetDescription () const`

get description

25.178.3.8 `unsigned short gdcm::Overlay::GetGroup () const`

Get Group number.

25.178.3.9 `const signed short* gdcm::Overlay::GetOrigin () const`

get origin

25.178.3.10 `const ByteValue& gdcm::Overlay::GetOverlayData () const`

25.178.3.11 `unsigned short gdcm::Overlay::GetRows () const`

get rows

25.178.3.12 `const char* gdcm::Overlay::GetType () const`

get type

25.178.3.13 `bool gdcm::Overlay::GetUnpackBuffer (unsigned char * buffer) const`

25.178.3.14 `bool gdcm::Overlay::GrabOverlayFromPixelData (DataSet const & ds)`

25.178.3.15 `bool gdcm::Overlay::IsEmpty () const`

25.178.3.16 `bool gdcm::Overlay::IsInPixelData () const`

25.178.3.17 `void gdcm::Overlay::IsInPixelData (bool b)`

25.178.3.18 `bool gdcm::Overlay::IsZero () const`

return true if all bits are set to 0

25.178.3.19 `void gdcm::Overlay::Print (std::ostream &) const` `[virtual]`

Print.

Reimplemented from `gdcm::Object`.

25.178.3.20 `void gdcm::Overlay::SetBitPosition (unsigned short bitposition)`

set bit position

25.178.3.21 `void gdcm::Overlay::SetBitsAllocated (unsigned short bitsallocated)`

set bits allocated

25.178.3.22 `void gdcm::Overlay::SetColumns (unsigned short columns)`

set columns

25.178.3.23 `void gdcm::Overlay::SetDescription (const char * description)`

set description

25.178.3.24 `void gdcm::Overlay::SetFrameOrigin (unsigned short frameorigin)`

set frame origin

25.178.3.25 `void gdcm::Overlay::SetGroup (unsigned short group)`

Set Group number.

25.178.3.26 void gdcm::Overlay::SetNumberOfFrames (unsigned int *numberofframes*)

set number of frames

25.178.3.27 void gdcm::Overlay::SetOrigin (const signed short * *origin*)

set origin

25.178.3.28 void gdcm::Overlay::SetOverlay (const char * *array*, unsigned int *length*)

set overlay from byte array + length

25.178.3.29 void gdcm::Overlay::SetRows (unsigned short *rows*)

set rows

25.178.3.30 void gdcm::Overlay::SetType (const char * *type*)

set type

25.178.3.31 void gdcm::Overlay::Update (const DataElement & *de*)

Update overlay from data element de:

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

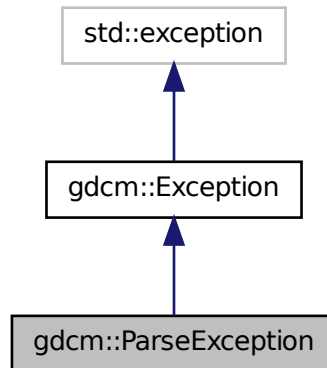
- gdcmOverlay.h

25.179 gdcm::ParseException Class Reference

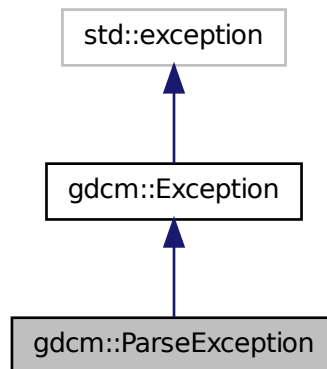
ParseException Standard exception handling object.

```
#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)`

25.179.1 Detailed Description

ParseException Standard exception handling object.

25.179.2 Constructor & Destructor Documentation

25.179.2.1 gdcM::ParseException::ParseException () [inline]

25.179.2.2 virtual gdcM::ParseException::~~ParseException () throw () [inline],[virtual]

25.179.3 Member Function Documentation

25.179.3.1 const DataElement& gdcM::ParseException::GetLastElement () const [inline]

25.179.3.2 ParseException& gdcM::ParseException::operator= (const ParseException & orig) [inline]

Assignment operator.

25.179.3.3 void gdcM::ParseException::SetLastElement (DataElement & de) [inline]

Equivalence operator.

Referenced by gdcM::Fragment::ReadValue().

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

- gdcMParseException.h

25.180 gdcM::Parser Class Reference

Parser ala XML_Parser from expat (SAX)

```
#include <gdcMParser.h>
```

Public Types

- typedef void(* EndElementHandler)(void *userData, const Tag &name)
- enum ErrorType {
 NoError,
 NoMemoryError,
 SyntaxError,
 NoElementsError,
 TagMismatchError,
 DuplicateAttributeError,
 JunkAfterDocElementError,
 UndefinedEntityError,
 UnexpectedStateError }
- typedef void(* StartElementHandler)(void *userData, const Tag &tag, const char *atts[])

Public Member Functions

- Parser ()
- ~Parser ()
- unsigned long GetCurrentByteIndex () const
- ErrorType GetErrorCode () const
- void * GetUserData () const
- bool Parse (const char *s, int len, bool isFinal)
- void SetElementHandler (StartElementHandler start, EndElementHandler end)
- void SetUserData (void *userData)

Static Public Member Functions

- static const char * GetErrorString (ErrorType const &err)

Protected Member Functions

- char * GetBuffer (int len)
- bool ParseBuffer (int len, bool isFinal)
- ErrorType Process ()

25.180.1 Detailed Description

Parser ala XML_Parser from expat (SAX)

Detailed description here

Note

Simple API for DICOM

25.180.2 Member Typedef Documentation

25.180.2.1 typedef void(* gdcmm::Parser::EndElementHandler)(void *userData, const Tag &name)

25.180.2.2 typedef void(* gdcmm::Parser::StartElementHandler)(void *userData, const Tag &tag, const char *atts[])

25.180.3 Member Enumeration Documentation

25.180.3.1 enum gdcmm::Parser::ErrorType

Enumerator:

NoError

NoMemoryError

SyntaxError

NoElementsError

TagMismatchError

DuplicateAttributeError

*JunkAfterDocElementError**UndefinedEntityError**UnexpectedStateError*

25.180.4 Constructor & Destructor Documentation

25.180.4.1 `gdcm::Parser::Parser () [inline]`25.180.4.2 `gdcm::Parser::~~Parser () [inline]`

25.180.5 Member Function Documentation

25.180.5.1 `char* gdcm::Parser::GetBuffer (int len) [protected]`25.180.5.2 `unsigned long gdcm::Parser::GetCurrentByteIndex () const`25.180.5.3 `ErrorType gdcm::Parser::GetErrorCode () const`25.180.5.4 `static const char* gdcm::Parser::GetErrorString (ErrorType const & err) [static]`25.180.5.5 `void* gdcm::Parser::GetUserData () const`25.180.5.6 `bool gdcm::Parser::Parse (const char * s, int len, bool isFinal)`25.180.5.7 `bool gdcm::Parser::ParseBuffer (int len, bool isFinal) [protected]`25.180.5.8 `ErrorType gdcm::Parser::Process () [protected]`25.180.5.9 `void gdcm::Parser::SetElementHandler (StartElementHandler start, EndElementHandler end)`25.180.5.10 `void gdcm::Parser::SetUserData (void * userData)`

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

- `gdcmParser.h`

25.181 gdcm::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

```
#include <gdcmPatient.h>
```

Public Member Functions

- `Patient ()`

25.181.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

25.181.2 Constructor & Destructor Documentation

25.181.2.1 `gdcm::Patient::Patient ()` `[inline]`

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

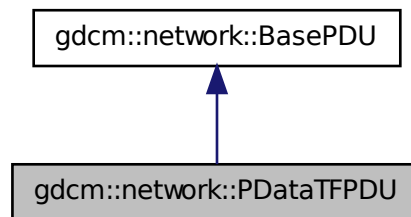
- `gdcmPatient.h`

25.182 `gdcm::network::PDataTFPDU` Class Reference

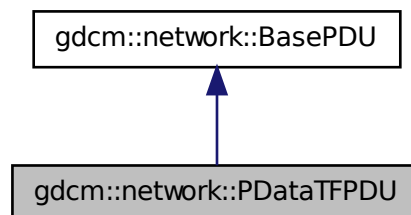
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

```
#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)

25.182.1 Detailed Description

PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

25.182.2 Member Typedef Documentation

25.182.2.1 `typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType`

25.182.3 Constructor & Destructor Documentation

25.182.3.1 `gdcm::network::PDataTFPDU::PDataTFPDU ()`

25.182.4 Member Function Documentation

25.182.4.1 `void gdcm::network::PDataTFPDU::AddPresentationDataValue (PresentationDataValue const & pdv)`
[inline]

25.182.4.2 `SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues () const` [inline]

25.182.4.3 `PresentationDataValue const& gdcm::network::PDataTFPDU::GetPresentationDataValue (SizeType i) const`
[inline]

25.182.4.4 `bool gdcm::network::PDataTFPDU::IsLastFragment () const` [virtual]

Implements `gdcm::network::BasePDU`.

25.182.4.5 `void gdcm::network::PDataTFPDU::Print (std::ostream & os) const` [virtual]

Implements `gdcm::network::BasePDU`.

25.182.4.6 `std::istream& gdcm::network::PDataTFPDU::Read (std::istream & is)` [virtual]

Implements `gdcm::network::BasePDU`.

25.182.4.7 `std::istream& gdcm::network::PDataTFPDU::ReadInto (std::istream & is, std::ostream & os)` [protected]

25.182.4.8 `size_t gdcm::network::PDataTFPDU::Size () const` [virtual]

Implements `gdcm::network::BasePDU`.

25.182.4.9 `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:

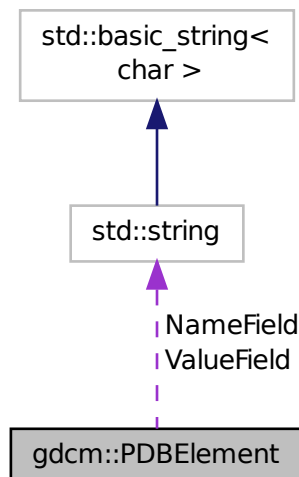
- `gdcmPDataTFPDU.h`

25.183 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)

25.183.1 Detailed Description

Class to represent a PDB Element.

See also

PDBHeader

25.183.2 Constructor & Destructor Documentation

25.183.2.1 gdcm::PDBElement::PDBElement () *[inline]*

25.183.3 Member Function Documentation

25.183.3.1 const char* gdcm::PDBElement::GetName () const *[inline]*

Set/Get Name.

25.183.3.2 const char* gdcm::PDBElement::GetValue () const *[inline]*

Set/Get Value.

25.183.3.3 bool gdcm::PDBElement::operator== (const PDBElement & de) const *[inline]*

References NameField, and ValueField.

25.183.3.4 void gdcmm::PDBelement::SetName (const char * name) [inline]

25.183.3.5 void gdcmm::PDBelement::SetValue (const char * value) [inline]

25.183.4 Friends And Related Function Documentation

25.183.4.1 std::ostream& operator<< (std::ostream & os, const PDBelement & val) [friend]

25.183.5 Member Data Documentation

25.183.5.1 std::string gdcmm::PDBelement::NameField [protected]

Referenced by gdcmm::operator<<(), and operator==().

25.183.5.2 std::string gdcmm::PDBelement::ValueField [protected]

Referenced by gdcmm::operator<<(), and operator==().

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

- gdcmmPDBelement.h

25.184 gdcmm::PDBHeader Class Reference

Class for PDBHeader.

```
#include <gdcmmPDBHeader.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)`

25.184.1 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

25.184.2 Constructor & Destructor Documentation

25.184.2.1 `gdcm::PDBHeader::PDBHeader ()` `[inline]`

25.184.2.2 `gdcm::PDBHeader::~~PDBHeader ()` `[inline]`

25.184.3 Member Function Documentation

25.184.3.1 `bool gdcm::PDBHeader::FindPDBElementByName (const char * name)`

Return true if the PDB element matching name is found or not.

25.184.3.2 `const PDBElement& gdcm::PDBHeader::GetPDBEEnd () const` `[protected]`

25.184.3.3 `const PDBElement& gdcm::PDBHeader::GetPDBElementByName (const char * name)`

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

25.184.3.4 `static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag ()` `[static]`

Return the Private Tag where the PDB header is stored within a DICOM DataSet.

25.184.3.5 `bool gdcm::PDBHeader::LoadFromDataElement (DataElement const & de)`

Load the PDB Header from a DataElement of a DataSet.

25.184.3.6 void gdcmm::PDBHeader::Print (std::ostream & os) const

Print.

Referenced by gdcmm::operator<<().

25.184.4 Friends And Related Function Documentation

25.184.4.1 std::ostream& operator<< (std::ostream & os, const PDBHeader & d) [friend]

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

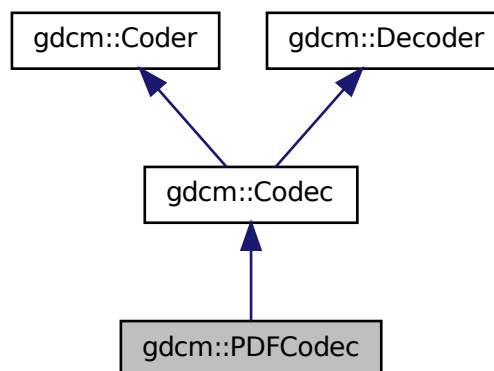
- gdcmmPDBHeader.h

25.185 gdcmm::PDFCodec Class Reference

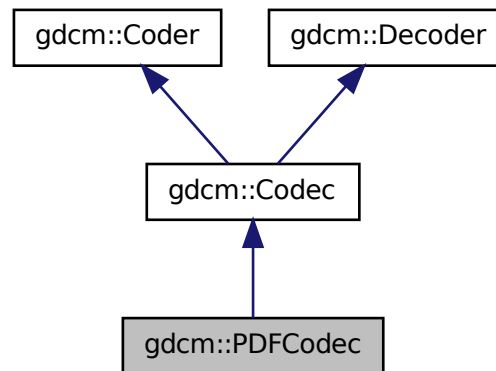
PDFCodec class.

```
#include <gdcmmPDFCodec.h>
```

Inheritance diagram for gdcmm::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.

25.185.1 Detailed Description

PDFCodec class.

25.185.2 Constructor & Destructor Documentation

25.185.2.1 gdcm::PDFCodec::PDFCodec ()

25.185.2.2 gdcm::PDFCodec::~~PDFCodec ()

25.185.3 Member Function Documentation

25.185.3.1 bool gdcm::PDFCodec::CanCode (TransferSyntax const &) const [inline],[virtual]

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

25.185.3.2 `bool gdcmm::PDFCodec::CanDecode (TransferSyntax const &) const` `[inline],[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcmm::Decoder.

25.185.3.3 `bool gdcmm::PDFCodec::Decode (DataElement const & is, DataElement & os)` `[virtual]`

Decode.

Reimplemented from gdcmm::Decoder.

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

- gdcmmPDFCodec.h

25.186 gdcmm::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 <gdcmmPDUFactory.h>
```

Static Public Member Functions

- static BasePDU * ConstructAbortPDU ()
- static BasePDU * ConstructPDU (uint8_t itemtype)
- static BasePDU * ConstructReleasePDU ()
- static std::vector< BasePDU * > CreateCEchoPDU (const ULConnection &inConnection)
- static std::vector< BasePDU * > CreateCFindPDU (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector< BasePDU * > CreateCMovePDU (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector< BasePDU * > CreateCStoreRQPDU (const ULConnection &inConnection, const File &file)
- static std::vector< BasePDU * > CreateCStoreRSPPDU (const DataSet *inDataSet, const BasePDU *inPC)
- static EEventID DetermineEventByPDU (const BasePDU *inPDU)
- static std::vector< PresentationDataValue > GetPDVs (const std::vector< BasePDU * > &inDataPDUs)

25.186.1 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.

25.186.2 Member Function Documentation

25.186.2.1 `static BasePDU* gdcmm::network::PDUFactory::ConstructAbortPDU ()` `[static]`

25.186.2.2 `static BasePDU* gdcmm::network::PDUFactory::ConstructPDU (uint8_t itemtype)` `[static]`

- 25.186.2.3 static **BasePDU*** gdcm::network::PDUFactory::ConstructReleasePDU () [static]
- 25.186.2.4 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCEchoPDU (const ULConnection & *inConnection*) [static]
- 25.186.2.5 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCFindPDU (const ULConnection & *inConnection*, const **BaseRootQuery** * *inRootQuery*) [static]
- 25.186.2.6 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCMovePDU (const ULConnection & *inConnection*, const **BaseRootQuery** * *inRootQuery*) [static]
- 25.186.2.7 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCStoreRQPDU (const ULConnection & *inConnection*, const **File** & *file*) [static]
- 25.186.2.8 static std::vector<**BasePDU***> gdcm::network::PDUFactory::CreateCStoreRSPPDU (const **DataSet** * *inDataSet*, const **BasePDU** * *inPC*) [static]
- 25.186.2.9 static **EEventID** gdcm::network::PDUFactory::DetermineEventByPDU (const **BasePDU** * *inPDU*) [static]
- 25.186.2.10 static std::vector<**PresentationDataValue**> gdcm::network::PDUFactory::GetPDVs (const std::vector<**BasePDU** * > & *inDataPDUs*) [static]

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

- gdcmPDUFactory.h

25.187 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 = '^'

25.187.1 Detailed Description

PersonName class.

25.187.2 Member Function Documentation

25.187.2.1 unsigned int gdcm::PersonName::GetMaxLength () const [inline]

25.187.2.2 unsigned int gdcm::PersonName::GetNumberOfComponents () const [inline]

25.187.2.3 void gdcm::PersonName::Print (std::ostream & os) const [inline]

25.187.2.4 void gdcm::PersonName::SetBlob (const std::vector< char > & v) [inline]

25.187.2.5 void gdcm::PersonName::SetComponents (const char * comp1 = "", const char * comp2 = "", const char * comp3 = "", const char * comp4 = "", const char * comp5 = "") [inline]

25.187.2.6 void gdcm::PersonName::SetComponents (const char * components[]) [inline]

25.187.3 Member Data Documentation

25.187.3.1 char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]

25.187.3.2 const unsigned int gdcm::PersonName::MaxLength = 64 [static]

25.187.3.3 const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5 [static]

25.187.3.4 const char gdcm::PersonName::Padding = '' [static]

25.187.3.5 const char gdcm::PersonName::Separator = '^' [static]

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

- gdcmPersonName.h

25.188 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
- PType GetType () const
- bool IsLossless () const
- bool IsLossy () const
- bool IsSameColorSpace (PhotometricInterpretation const &pi) const
- operator PType () const

Static Public Member Functions

- static const char * GetPIString (PType pi)
- static PType GetPType (const char *pi)
- static bool IsRetired (PType pi)

Friends

- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &pi)

25.188.1 Detailed Description

Class to represent an PhotometricInterpretation.

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, HelloVizWorld.cxx, and iU22tomultisc.cxx.

25.188.2 Member Enumeration Documentation

25.188.2.1 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

25.188.3 Constructor & Destructor Documentation

25.188.3.1 gdcm::PhotometricInterpretation::PhotometricInterpretation (PIType *pi* = UNKNOWN) [inline]

25.188.4 Member Function Documentation

25.188.4.1 static const char* gdcm::PhotometricInterpretation::GetPIString (PIType *pi*) [static]

Referenced by gdcm::operator<<().

25.188.4.2 static PIType gdcm::PhotometricInterpretation::GetPIType (const char * *pi*) [static]

25.188.4.3 unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel () const

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

25.188.4.4 const char* gdcm::PhotometricInterpretation::GetString () const

25.188.4.5 PIType gdcm::PhotometricInterpretation::GetType () const [inline]

25.188.4.6 bool gdcm::PhotometricInterpretation::IsLossless () const

25.188.4.7 bool gdcm::PhotometricInterpretation::IsLossy () const

25.188.4.8 static bool gdcm::PhotometricInterpretation::IsRetired (PIType *pi*) [static]

25.188.4.9 `bool gdcm::PhotometricInterpretation::IsSameColorSpace (PhotometricInterpretation const & pi) const`

25.188.4.10 `gdcm::PhotometricInterpretation::operator PType () const` `[inline]`

25.188.5 Friends And Related Function Documentation

25.188.5.1 `std::ostream& operator<< (std::ostream & os, const PhotometricInterpretation & pi)` `[friend]`

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

- `gdcmPhotometricInterpretation.h`

25.189 gdcm::PixelFormat Class Reference

PixelFormat.

```
#include <gdcmPixelFormat.h>
```

Public Types

- enum ScalarType {
 UINT8,
 INT8,
 UINT12,
 INT12,
 UINT16,
 INT16,
 UINT32,
 INT32,
 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)
- ~PixelFormat ()
- 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 IsValid ()`
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)`

25.189.1 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

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, iU22tomultisc.cxx, and threadgdcmm.cxx.

25.189.2 Member Enumeration Documentation

25.189.2.1 enum gdcmm::PixelFormat::ScalarType

Enumerator:

UINT8
INT8
UINT12
INT12
UINT16
INT16
UINT32
INT32
FLOAT16
FLOAT32
FLOAT64
SINGLEBIT
UNKNOWN

25.189.3 Constructor & Destructor Documentation

25.189.3.1 gdcmm::PixelFormat::PixelFormat (unsigned short *samplesperpixel* = 1, unsigned short *bitsallocated* = 8, unsigned short *bitsstored* = 8, unsigned short *highbit* = 7, unsigned short *pixelrepresentation* = 0) [inline], [explicit]

25.189.3.2 gdcmm::PixelFormat::PixelFormat (ScalarType *st*)

25.189.3.3 gdcmm::PixelFormat::~~PixelFormat () [inline]

25.189.4 Member Function Documentation

25.189.4.1 unsigned short gdcmm::PixelFormat::GetBitsAllocated () const [inline]

BitsAllocated see Tag (0028,0100) US Bits Allocated.

Examples:

GetJPEGSamplePrecision.cxx.

25.189.4.2 unsigned short gdcmm::PixelFormat::GetBitsStored () const [inline]

BitsStored see Tag (0028,0101) US Bits Stored.

Examples:

GetJPEGSamplePrecision.cxx.

25.189.4.3 `unsigned short gdcm::PixelFormat::GetHighBit () const` `[inline]`

HighBit see Tag (0028,0102) US High Bit.

25.189.4.4 `int64_t gdcm::PixelFormat::GetMax () const`

return the max possible of the pixel

25.189.4.5 `int64_t gdcm::PixelFormat::GetMin () const`

return the min possible of the pixel

25.189.4.6 `unsigned short gdcm::PixelFormat::GetPixelRepresentation () const` `[inline]`

PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.

25.189.4.7 `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.
```

25.189.4.8 `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.
```

25.189.4.9 `ScalarType gdcm::PixelFormat::GetScalarType () const`

ScalarType does not take into account the sample per pixel.

25.189.4.10 `const char* gdcm::PixelFormat::GetScalarTypeAsString () const`

25.189.4.11 `bool gdcm::PixelFormat::IsValid ()`

return IsValid

- 25.189.4.12 `gdcm::PixelFormat::operator ScalarType () const` [inline]
- 25.189.4.13 `bool gdcm::PixelFormat::operator!= (ScalarType st) const` [inline]
- 25.189.4.14 `bool gdcm::PixelFormat::operator!= (const PixelFormat & pf) const` [inline]
- 25.189.4.15 `bool gdcm::PixelFormat::operator== (ScalarType st) const` [inline]
- 25.189.4.16 `bool gdcm::PixelFormat::operator== (const PixelFormat & pf) const` [inline]
- 25.189.4.17 `void gdcm::PixelFormat::Print (std::ostream & os) const`

Print.

Referenced by `gdcm::operator<<()`.

- 25.189.4.18 `void gdcm::PixelFormat::SetBitsAllocated (unsigned short ba)` [inline]
- 25.189.4.19 `void gdcm::PixelFormat::SetBitsStored (unsigned short bs)` [inline]
- 25.189.4.20 `void gdcm::PixelFormat::SetHighBit (unsigned short hb)` [inline]
- 25.189.4.21 `void gdcm::PixelFormat::SetPixelRepresentation (unsigned short pr)` [inline]
- 25.189.4.22 `void gdcm::PixelFormat::SetSamplesPerPixel (unsigned short spp)` [inline]

Examples:

`CreateARGBImage.cxx`, `CreateCMYKImage.cxx`, and `GenFakelImage.cxx`.

References `gdcmAssertMacro`.

- 25.189.4.23 `void gdcm::PixelFormat::SetScalarType (ScalarType st)`

Set PixelFormat based only on the ScalarType

Warning

: You need to call `SetScalarType` *before* `SetSamplesPerPixel`

- 25.189.4.24 `bool gdcm::PixelFormat::Validate ()` [protected]

When image with 24/24/23 was read, need to validate.

Referenced by `gdcm::Bitmap::SetPixelFormat()`.

25.189.5 Friends And Related Function Documentation

- 25.189.5.1 `friend class Bitmap` [friend]

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

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

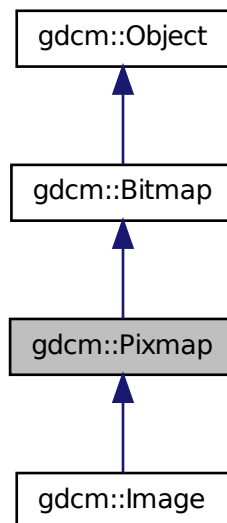
- `gdcmPixelFormat.h`

25.190 gdcm::Pixmap Class Reference

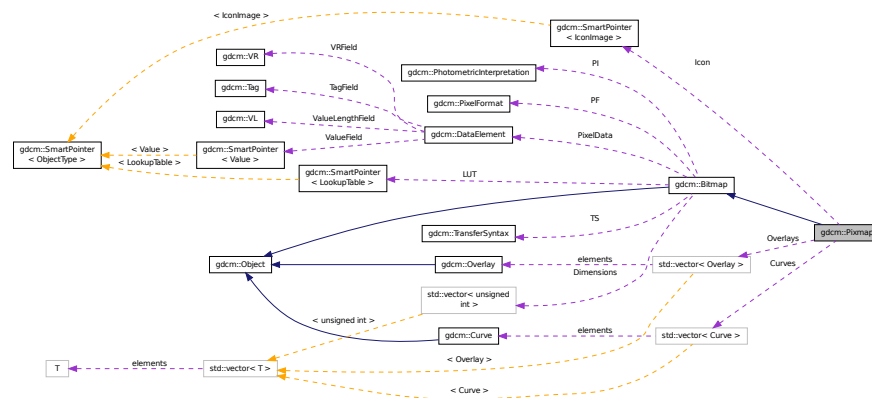
Pixmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

```
#include <gdcmPixmap.h>
```

Inheritance diagram for `gdcm::Pixmap`:



Collaboration diagram for gdcm::Pixmap:



Public Member Functions

- Pixmap ()
- ~Pixmap ()
- bool AreOverlaysInPixelData () const
returns if Overlays are stored in the unused bit of the pixel data:
- Curve & GetCurve (unsigned int i=0)
Curve: group 50xx.
- const Curve & GetCurve (unsigned int 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

25.190.1 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

25.190.2 Constructor & Destructor Documentation

25.190.2.1 `gdcm::Pixmap::Pixmap ()`

25.190.2.2 `gdcm::Pixmap::~~Pixmap ()`

25.190.3 Member Function Documentation

25.190.3.1 `bool gdcm::Pixmap::AreOverlaysInPixelData () const` [virtual]

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from `gdcm::Bitmap`.

25.190.3.2 `Curve& gdcm::Pixmap::GetCurve (unsigned int i = 0)` [inline]

Curve: group 50xx.

25.190.3.3 `const Curve& gdcm::Pixmap::GetCurve (unsigned int i = 0) const` [inline]

25.190.3.4 `const IconImage& gdcm::Pixmap::GetIconImage () const` [inline]

Set/Get Icon Image.

25.190.3.5 `IconImage& gdcm::Pixmap::GetIconImage ()` [inline]

25.190.3.6 `size_t gdcm::Pixmap::GetNumberOfCurves () const` [inline]

25.190.3.7 `size_t gdcm::Pixmap::GetNumberOfOverlays () const` [inline]

25.190.3.8 `Overlay& gdcm::Pixmap::GetOverlay (size_t i = 0)` [inline]

Overlay: group 60xx.

25.190.3.9 `const Overlay& gdcm::Pixmap::GetOverlay (size_t i = 0) const` [inline]

25.190.3.10 `void gdcm::Pixmap::Print (std::ostream &) const` [virtual]

Reimplemented from `gdcm::Bitmap`.

Reimplemented in gdcm::Image.

25.190.3.11 void gdcm::Pixmap::RemoveOverlay (size_t *i*) [inline]

25.190.3.12 void gdcm::Pixmap::SetIconImage (IconImage const & *ii*) [inline]

25.190.3.13 void gdcm::Pixmap::SetNumberOfCurves (size_t *n*) [inline]

25.190.3.14 void gdcm::Pixmap::SetNumberOfOverlays (size_t *n*) [inline]

25.190.4 Member Data Documentation

25.190.4.1 std::vector<Curve> gdcm::Pixmap::Curves [protected]

25.190.4.2 SmartPointer<IconImage> gdcm::Pixmap::Icon [protected]

25.190.4.3 std::vector<Overlay> gdcm::Pixmap::Overlays [protected]

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

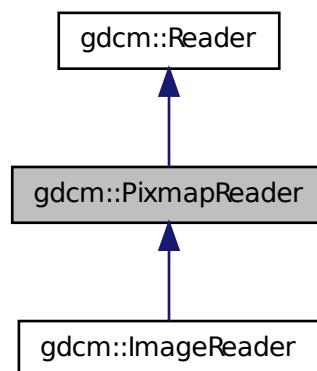
- gdcmPixmap.h

25.191 gdcm::PixmapReader Class Reference

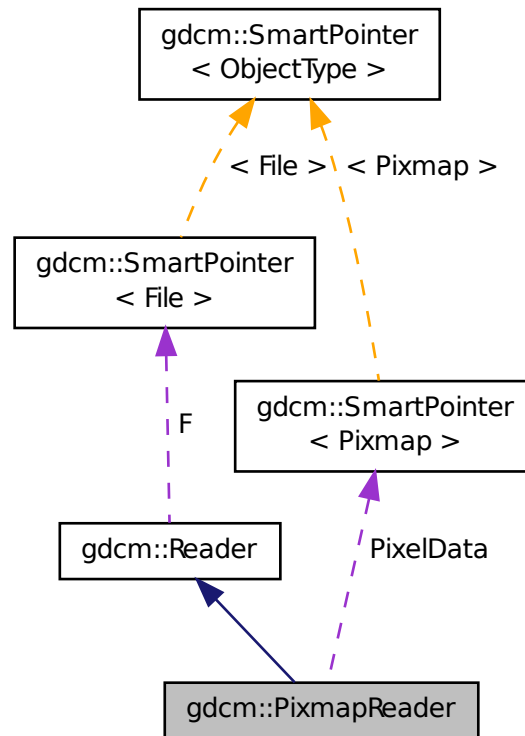
PixmapReader.

```
#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)`

Protected Attributes

- `SmartPointer< Pixmap > PixelData`

25.191.1 Detailed Description

PixmapReader.

Note

its role is to convert the DICOM DataSet into a gdcm::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

25.191.2 Constructor & Destructor Documentation

25.191.2.1 `gdcm::PixmapReader::PixmapReader ()`

25.191.2.2 `virtual gdcm::PixmapReader::~~PixmapReader () [virtual]`

25.191.3 Member Function Documentation

25.191.3.1 `const Pixmap& gdcm::PixmapReader::GetPixmap () const`

Return the read image (need to call Read() first)

25.191.3.2 `Pixmap& gdcm::PixmapReader::GetPixmap ()`

25.191.3.3 `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::ImageReader.

25.191.3.4 `virtual bool gdcm::PixmapReader::ReadACRNEMAIImage () [protected], [virtual]`

Reimplemented in gdcm::ImageReader.

25.191.3.5 `virtual bool gdcm::PixmapReader::ReadImage (MediaStorage const & ms) [protected], [virtual]`

Reimplemented in gdcm::ImageReader.

25.191.4 Member Data Documentation

25.191.4.1 `SmartPointer<Pixmap> gdcM::PixmapReader::PixelData` [protected]

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

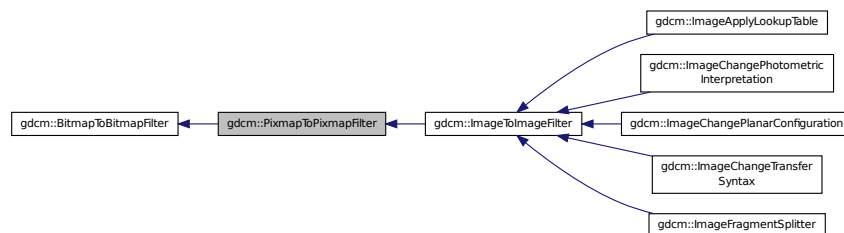
- `gdcMPixmapReader.h`

25.192 `gdcM::PixmapToPixmapFilter` Class Reference

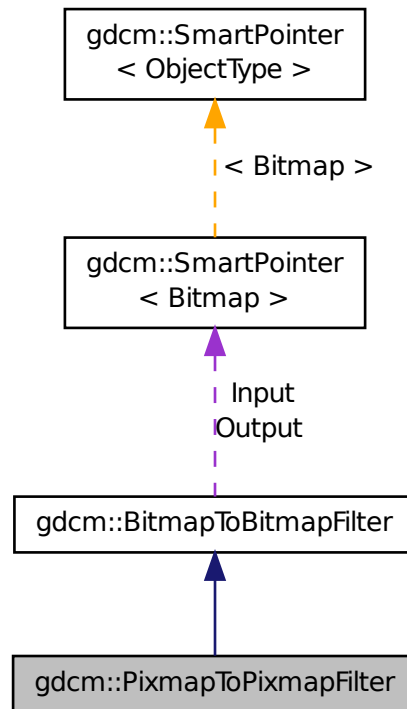
`PixmapToPixmapFilter` class Super class for all filter taking an image and producing an output image.

```
#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.

Additional Inherited Members

25.192.1 Detailed Description

`PixmapToPixmapFilter` class Super class for all filter taking an image and producing an output image.

25.192.2 Constructor & Destructor Documentation

25.192.2.1 `gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ()`

25.192.2.2 `gdcm::PixmapToPixmapFilter::~~PixmapToPixmapFilter ()` `[inline]`

25.192.3 Member Function Documentation

25.192.3.1 `Pixmap& gdcm::PixmapToPixmapFilter::GetInput ()`

Reimplemented in `gdcm::ImageToImageFilter`.

25.192.3.2 `const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput () const`

Get Output image.

Reimplemented from `gdcm::BitmapToBitmapFilter`.

Reimplemented in `gdcm::ImageToImageFilter`.

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

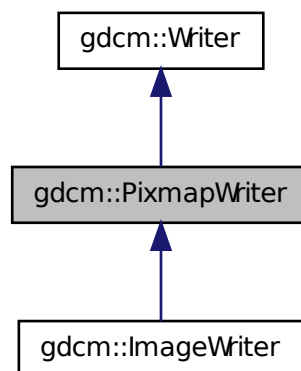
- `gdcmPixmapToPixmapFilter.h`

25.193 gdcm::PixmapWriter Class Reference

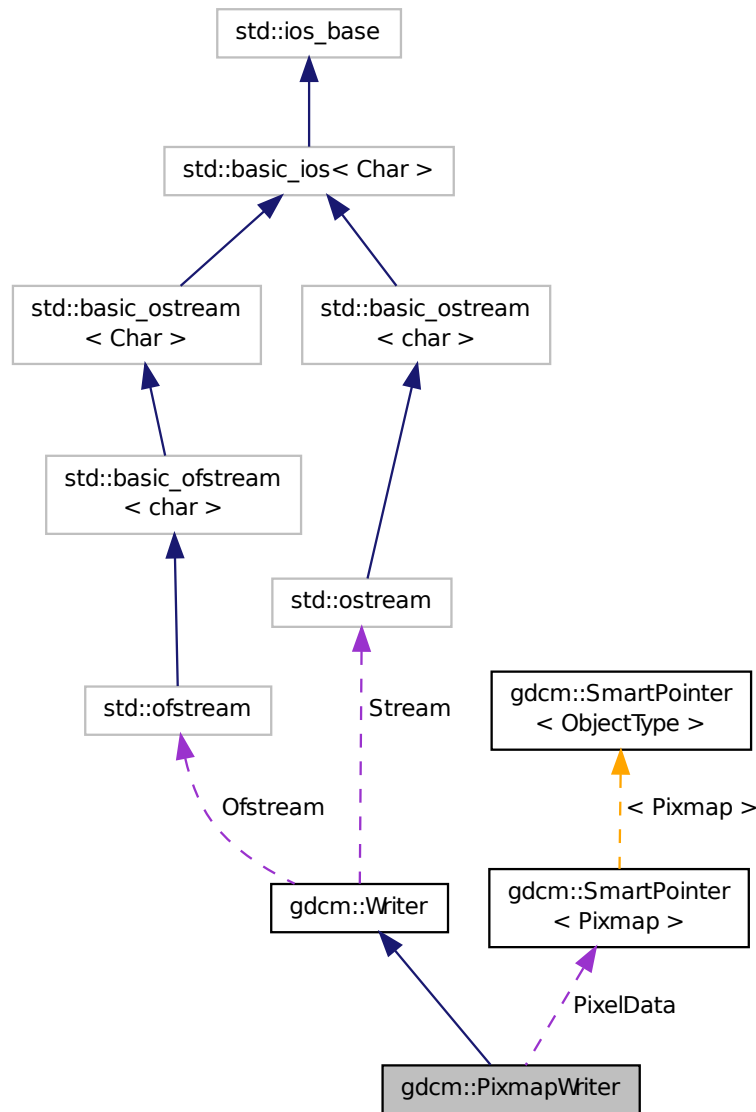
`PixmapWriter` This class will takes two inputs:

```
#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 ()

Protected Attributes

- SmartPointer< Pixmap > PixelData

25.193.1 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)

25.193.2 Constructor & Destructor Documentation

25.193.2.1 `gdcm::PixmapWriter::PixmapWriter ()`

25.193.2.2 `gdcm::PixmapWriter::~~PixmapWriter ()`

25.193.3 Member Function Documentation

25.193.3.1 `void gdcm::PixmapWriter::DolconImage (DataSet & ds, Pixmap const & image)` `[protected]`

25.193.3.2 `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`.

25.193.3.3 `virtual Pixmap& gdcm::PixmapWriter::GetImage ()` `[inline],[virtual]`

Reimplemented in `gdcm::ImageWriter`.

25.193.3.4 `const Pixmap& gdcm::PixmapWriter::GetPixmap () const` [inline]

25.193.3.5 `Pixmap& gdcm::PixmapWriter::GetPixmap ()` [inline]

25.193.3.6 `bool gdcm::PixmapWriter::PrepareWrite ()` [protected]

25.193.3.7 `virtual void gdcm::PixmapWriter::SetImage (Pixmap const & img)` [virtual]

Examples:

CompressImage.cxx, GenFakelImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, and MergeTwoFiles.cxx.

25.193.3.8 `void gdcm::PixmapWriter::SetPixmap (Pixmap const & img)`

25.193.3.9 `bool gdcm::PixmapWriter::Write ()` [virtual]

Write.

Reimplemented from `gdcm::Writer`.

Reimplemented in `gdcm::ImageWriter`.

25.193.4 Member Data Documentation

25.193.4.1 `SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData` [protected]

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

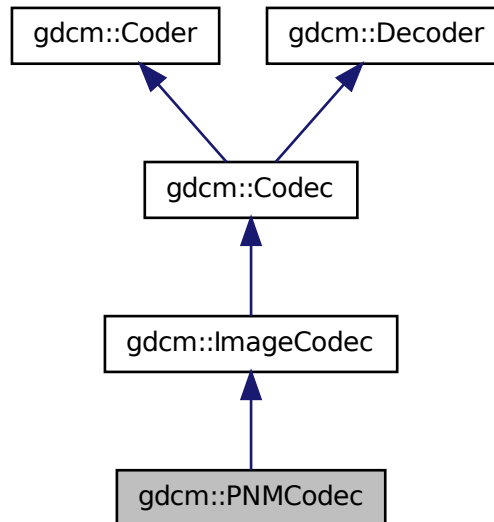
- `gdcmPixmapWriter.h`

25.194 gdcm::PNMCodec Class Reference

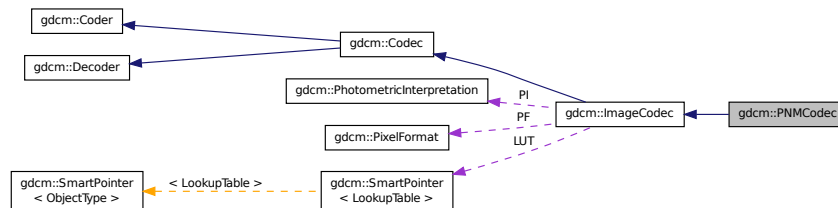
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.

```
#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)
- `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

25.194.1 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.

25.194.2 Constructor & Destructor Documentation

25.194.2.1 `gdcm::PNMCodec::PNMCodec ()`

25.194.2.2 `gdcm::PNMCodec::~~PNMCodec ()`

25.194.3 Member Function Documentation

25.194.3.1 `bool gdcm::PNMCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from `gdcm::ImageCodec`.

25.194.3.2 `bool gdcm::PNMCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from `gdcm::ImageCodec`.

25.194.3.3 `unsigned long gdcm::PNMCodec::GetBufferLength () const` [inline]

25.194.3.4 `bool gdcm::PNMCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from `gdcm::ImageCodec`.

25.194.3.5 `bool gdcm::PNMCodec::Read (const char * filename, DataElement & out) const`

25.194.3.6 `void gdcm::PNMCodec::SetBufferLength (unsigned long l)` [inline]

25.194.3.7 `bool gdcm::PNMCodec::Write (const char * filename, const DataElement & out) const`

Examples:

ExtractIconFromFile.cxx.

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

- gdcmPidCodec.h

25.195 gdcmPid::Preamble Class Reference

DICOM Preamble (Part 10)

```
#include <gdcmPidPreamble.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)

25.195.1 Detailed Description

DICOM Preamble (Part 10)

25.195.2 Constructor & Destructor Documentation

25.195.2.1 gdcmPid::Preamble::Preamble ()

25.195.2.2 gdcmPid::Preamble::~~Preamble ()

25.195.2.3 gdcmPid::Preamble::Preamble (Preamble const &) [inline]

25.195.3 Member Function Documentation

- 25.195.3.1 void gdcm::Preamble::Clear ()
- 25.195.3.2 void gdcm::Preamble::Create ()
- 25.195.3.3 const char* gdcm::Preamble::GetInternal () const [inline]
- 25.195.3.4 VL gdcm::Preamble::GetLength () const [inline]
- 25.195.3.5 bool gdcm::Preamble::IsEmpty () const [inline]
- 25.195.3.6 bool gdcm::Preamble::IsValid () const [inline],[protected]
- 25.195.3.7 Preamble& gdcm::Preamble::operator= (Preamble const &) [inline]
- 25.195.3.8 void gdcm::Preamble::Print (std::ostream & os) const
- 25.195.3.9 std::istream& gdcm::Preamble::Read (std::istream & is)
- 25.195.3.10 void gdcm::Preamble::Remove ()
- 25.195.3.11 void gdcm::Preamble::Valid ()
- 25.195.3.12 std::ostream const& gdcm::Preamble::Write (std::ostream & os) const

25.195.4 Friends And Related Function Documentation

- 25.195.4.1 std::ostream& operator<< (std::ostream & _os, const Preamble & _val) [friend]

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

- gdcmPreamble.h

25.196 gdcm::PresentationContext Class Reference

PresentationContext.

```
#include <gdcmPresentationContext.h>
```

Public Types

- typedef
TransferSyntaxArrayType::size_type SizeType
- typedef std::vector< std::string > TransferSyntaxArrayType

Public Member Functions

- PresentationContext ()

- PresentationContext (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
- void AddTransferSyntax (const char *tsstr)
- const char * GetAbstractSyntax () const
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- const char * GetTransferSyntax (SizeType i) const
- bool operator== (const PresentationContext &pc) const
- void Print (std::ostream &os) const
- void SetAbstractSyntax (const char *as)
- void SetPresentationContextID (uint8_t id)

25.196.1 Detailed Description

PresentationContext.

See also

PresentationContextAC PresentationContextRQ

25.196.2 Member Typedef Documentation

25.196.2.1 `typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType`

25.196.2.2 `typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType`

25.196.3 Constructor & Destructor Documentation

25.196.3.1 `gdcm::PresentationContext::PresentationContext ()`

25.196.3.2 `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).

25.196.4 Member Function Documentation

25.196.4.1 `void gdcm::PresentationContext::AddTransferSyntax (const char * tsstr)`

25.196.4.2 `const char* gdcm::PresentationContext::GetAbstractSyntax () const` `[inline]`

25.196.4.3 `SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes () const` `[inline]`

25.196.4.4 `uint8_t gdcm::PresentationContext::GetPresentationContextID () const`

25.196.4.5 `const char* gdcm::PresentationContext::GetTransferSyntax (SizeType i) const` `[inline]`

25.196.4.6 `bool gdcm::PresentationContext::operator== (const PresentationContext & pc) const` `[inline]`

25.196.4.7 void gdcm::PresentationContext::Print (std::ostream & os) const

25.196.4.8 void gdcm::PresentationContext::SetAbstractSyntax (const char * as) [inline]

25.196.4.9 void gdcm::PresentationContext::SetPresentationContextID (uint8_t id)

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

- gdcmPresentationContext.h

25.197 gdcm::network::PresentationContextAC Class Reference

PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

```
#include <gdcmPresentationContextAC.h>
```

Public Member Functions

- PresentationContextAC ()
- uint8_t GetPresentationContextID () const
- TransferSyntaxSub const & GetTransferSyntax () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetPresentationContextID (uint8_t id)
- void SetTransferSyntax (TransferSyntaxSub const &ts)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.197.1 Detailed Description

PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

See also

PresentationContext

25.197.2 Constructor & Destructor Documentation

25.197.2.1 gdcm::network::PresentationContextAC::PresentationContextAC ()

25.197.3 Member Function Documentation

25.197.3.1 uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID () const [inline]

25.197.3.2 TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax () const [inline]

25.197.3.3 void gdcm::network::PresentationContextAC::Print (std::ostream & os) const

25.197.3.4 std::istream& gdcm::network::PresentationContextAC::Read (std::istream & is)

25.197.3.5 void gdcmm::network::PresentationContextAC::SetPresentationContextID (uint8_t id)

25.197.3.6 void gdcmm::network::PresentationContextAC::SetTransferSyntax (TransferSyntaxSub const & ts)

25.197.3.7 size_t gdcmm::network::PresentationContextAC::Size () const

25.197.3.8 const std::ostream& gdcmm::network::PresentationContextAC::Write (std::ostream & os) const

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

- gdcmmPresentationContextAC.h

25.198 gdcmm::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 <gdcmmPresentationContextGenerator.h>
```

Public Types

- typedef std::vector
< PresentationContext > PresentationContextArrayType
- typedef
PresentationContextArrayType::size_type SizeType

Public Member Functions

- PresentationContextGenerator ()
- 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 *as, const char *ts)
- const char * GetDefaultTransferSyntax () const

25.198.1 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.

25.198.2 Member Typedef Documentation

25.198.2.1 **typedef std::vector<PresentationContext> gdcmm::PresentationContextGenerator::PresentationContextArrayType**

25.198.2.2 **typedef PresentationContextArrayType::size_type gdcmm::PresentationContextGenerator::SizeType**

25.198.3 Constructor & Destructor Documentation

25.198.3.1 **gdcmm::PresentationContextGenerator::PresentationContextGenerator ()**

25.198.4 Member Function Documentation

25.198.4.1 **bool gdcmm::PresentationContextGenerator::AddPresentationContext (const char * as, const char * ts)**
[protected]

25.198.4.2 **bool gdcmm::PresentationContextGenerator::GenerateFromFilenames (const Directory::FilenamesType & files)**

Generate the **PresentationContext** array from a File-Set. File specified needs to be valid DICOM files. Used for C-STORE operations

Examples:

CStoreQtProgress.cxx.

25.198.4.3 **bool gdcmm::PresentationContextGenerator::GenerateFromUID (UIDs::TSName asname)**

Generate the **PresentationContext** array from a UID (eg. VerificationSOPClass)

25.198.4.4 `const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax () const` [protected]

25.198.4.5 `PresentationContextArrayType const& gdcm::PresentationContextGenerator::GetPresentationContexts ()`
[inline]

Examples:

CStoreQtProgress.cxx.

25.198.4.6 `void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (const TransferSyntax & ts)`

Not implemented for now. GDCM internally uses Implicit Little Endian.

25.198.4.7 `void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ()`

25.198.4.8 `void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ()`

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

- gdcmPresentationContextGenerator.h

25.199 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 &as)
- void SetPresentationContextID (uint8_t id)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.199.1 Detailed Description

PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

See also

PresentationContextAC

25.199.2 Member Typedef Documentation

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

25.199.3 Constructor & Destructor Documentation

25.199.3.1 `gdcm::network::PresentationContextRQ::PresentationContextRQ ()`

25.199.3.2 `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).

25.199.3.3 `gdcm::network::PresentationContextRQ::PresentationContextRQ (const PresentationContext & pc)`

25.199.4 Member Function Documentation

25.199.4.1 `void gdcm::network::PresentationContextRQ::AddTransferSyntax (TransferSyntaxSub const & ts)`

25.199.4.2 `AbstractSyntax const& gdcm::network::PresentationContextRQ::GetAbstractSyntax () const` `[inline]`

25.199.4.3 `AbstractSyntax& gdcm::network::PresentationContextRQ::GetAbstractSyntax ()` `[inline]`

25.199.4.4 `SizeType gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes () const` `[inline]`

25.199.4.5 `uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID () const`

25.199.4.6 `TransferSyntaxSub const& gdcm::network::PresentationContextRQ::GetTransferSyntax (SizeType i) const` `[inline]`

25.199.4.7 `TransferSyntaxSub& gdcm::network::PresentationContextRQ::GetTransferSyntax (SizeType i)` `[inline]`

- 25.199.4.8 `std::vector<TransferSyntaxSub> const& gdcmm::network::PresentationContextRQ::GetTransferSyntaxes () const`
[inline]
- 25.199.4.9 `bool gdcmm::network::PresentationContextRQ::operator== (const PresentationContextRQ & pc) const`
[inline]
- 25.199.4.10 `void gdcmm::network::PresentationContextRQ::Print (std::ostream & os) const`
- 25.199.4.11 `std::istream& gdcmm::network::PresentationContextRQ::Read (std::istream & is)`
- 25.199.4.12 `void gdcmm::network::PresentationContextRQ::SetAbstractSyntax (AbstractSyntax const & as)`
- 25.199.4.13 `void gdcmm::network::PresentationContextRQ::SetPresentationContextID (uint8_t id)`
- 25.199.4.14 `size_t gdcmm::network::PresentationContextRQ::Size () const`
- 25.199.4.15 `const std::ostream& gdcmm::network::PresentationContextRQ::Write (std::ostream & os) const`

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

- gdcmmPresentationContextRQ.h

25.200 gdcmm::network::PresentationDataValue Class Reference

PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

```
#include <gdcmmPresentationDataValue.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)

25.200.1 Detailed Description

PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

25.200.2 Constructor & Destructor Documentation

25.200.2.1 `gdcm::network::PresentationDataValue::PresentationDataValue ()`

25.200.3 Member Function Documentation

25.200.3.1 `static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (const std::vector< PresentationDataValue > & inPDVs) [static]`

Warning

DataSet will be read as Implicit Little Endian TS

25.200.3.2 `const std::string& gdcm::network::PresentationDataValue::GetBlob () const`

25.200.3.3 `bool gdcm::network::PresentationDataValue::GetIsCommand () const`

25.200.3.4 `bool gdcm::network::PresentationDataValue::GetIsLastFragment () const`

25.200.3.5 `uint8_t gdcm::network::PresentationDataValue::GetMessageHeader () const [inline]`

25.200.3.6 `uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID () const [inline]`

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

25.200.3.8 `std::istream& gdcm::network::PresentationDataValue::Read (std::istream & is)`

25.200.3.9 `std::istream& gdcm::network::PresentationDataValue::ReadInto (std::istream & is, std::ostream & os)`

25.200.3.10 `void gdcm::network::PresentationDataValue::SetBlob (const std::string & partialblob)`

25.200.3.11 `void gdcm::network::PresentationDataValue::SetCommand (bool inCommand)`

25.200.3.12 `void gdcm::network::PresentationDataValue::SetDataSet (const DataSet & ds)`

Set DataSet. Write DataSet in implicit.

Warning

size of dataset should be below maxpdu size

25.200.3.13 `void gdcm::network::PresentationDataValue::SetLastFragment (bool inLast)`

25.200.3.14 `void gdcm::network::PresentationDataValue::SetMessageHeader (uint8_t messageheader) [inline]`

25.200.3.15 `void gdcmm::network::PresentationDataValue::SetPresentationContextID (uint8_t id) [inline]`

25.200.3.16 `size_t gdcmm::network::PresentationDataValue::Size () const`

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

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

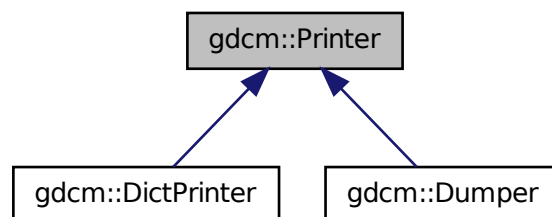
- `gdcmmPresentationDataValue.h`

25.201 gdcmm::Printer Class Reference

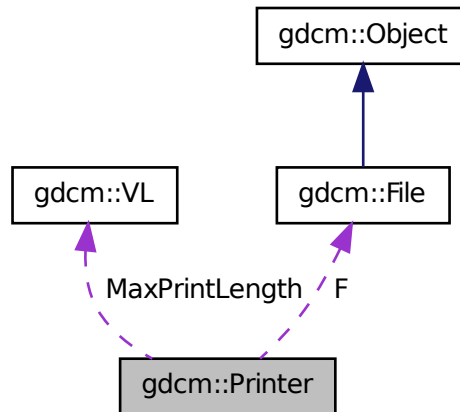
Printer class.

```
#include <gdcmmPrinter.h>
```

Inheritance diagram for `gdcmm::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::ostream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream &out, std::string const &indent)
- void PrintSQ (const SequenceOfItems *sqi, std::ostream &os, std::string const &indent)

Protected Attributes

- const File * F
- VL MaxPrintLength
- PrintStyles PrintStyle

25.201.1 Detailed Description

Printer class.

25.201.2 Member Enumeration Documentation

25.201.2.1 enum gdcm::Printer::PrintStyles

Enumerator:

VERBOSE_STYLE
CONDENSED_STYLE
XML

25.201.3 Constructor & Destructor Documentation

25.201.3.1 gdcm::Printer::Printer ()

25.201.3.2 gdcm::Printer::~~Printer ()

25.201.4 Member Function Documentation

25.201.4.1 PrintStyles gdcm::Printer::GetPrintStyle () const [inline]

Get PrintStyle value.

25.201.4.2 void gdcm::Printer::Print (std::ostream & os)

Print.

Reimplemented in gdcm::DictPrinter.

25.201.4.3 VR gdcm::Printer::PrintDataElement (std::ostream & os, const Dicts & dicts, const DataSet & ds, const DataElement & de, std::ostream & out, std::string const & indent) [protected]

25.201.4.4 void gdcm::Printer::PrintDataSet (const DataSet & ds, std::ostream & os, const std::string & s = " ")

Print an individual dataset.

25.201.4.5 void gdcm::Printer::PrintSQ (const SequenceOfItems * *sqi*, std::ostream & *os*, std::string const & *indent*)
[protected]

25.201.4.6 void gdcm::Printer::SetColor (bool *c*)

Set color mode or not.

25.201.4.7 void gdcm::Printer::SetFile (File const & *f*) [inline]

Set file.

25.201.4.8 void gdcm::Printer::SetStyle (PrintStyles *ps*) [inline]

Set PrintStyle value.

25.201.5 Member Data Documentation

25.201.5.1 const File* gdcm::Printer::F [protected]

25.201.5.2 VL gdcm::Printer::MaxPrintLength [protected]

25.201.5.3 PrintStyles gdcm::Printer::PrintStyle [protected]

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

- gdcmPrinter.h

25.202 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)`

25.202.1 Detailed Description

Private Dict.

25.202.2 Constructor & Destructor Documentation

25.202.2.1 `gdcmm::PrivateDict::PrivateDict ()` `[inline]`

25.202.2.2 `gdcmm::PrivateDict::~~PrivateDict ()` `[inline]`

25.202.3 Member Function Documentation

25.202.3.1 `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`.

25.202.3.2 `bool gdcmm::PrivateDict::FindDictEntry (const PrivateTag &tag) const` `[inline]`

25.202.3.3 `const DictEntry& gdcmm::PrivateDict::GetDictEntry (const PrivateTag &tag) const` `[inline]`

25.202.3.4 `bool gdcmm::PrivateDict::IsEmpty () const` `[inline]`

25.202.3.5 `void gdcmm::PrivateDict::LoadDefault ()` `[protected]`

25.202.3.6 `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()`.

25.202.3.7 `bool gdcmm::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.

25.202.4 Friends And Related Function Documentation

25.202.4.1 `friend class Dicts` `[friend]`

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

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

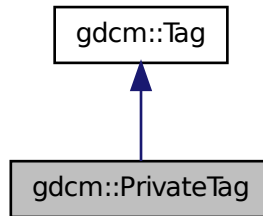
- `gdcmmDict.h`

25.203 gdcm::PrivateTag Class Reference

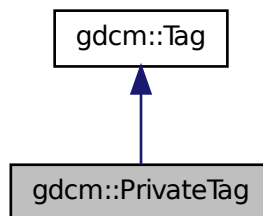
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

```
#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="")`
- `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)`

25.203.1 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:

csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, ELSCINT1WaveToText.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, PublicDict.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

25.203.2 Constructor & Destructor Documentation

25.203.2.1 `gdcm::PrivateTag::PrivateTag (uint16_t group = 0, uint16_t element = 0, const char * owner = " ") [inline]`

25.203.3 Member Function Documentation

25.203.3.1 `const char* gdcm::PrivateTag::GetOwner () const [inline]`

Examples:

PublicDict.cxx.

Referenced by operator<(), and gdcm::PrivateDict::PrintXML().

25.203.3.2 `bool gdcm::PrivateTag::operator< (const PrivateTag & _val) const [inline]`

References GetOwner(), and gdcm::System::StrCaseCmp().

25.203.3.3 `bool gdcm::PrivateTag::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

Reimplemented from gdcm::Tag.

25.203.3.4 `void gdcm::PrivateTag::SetOwner (const char * owner) [inline]`

25.203.4 Friends And Related Function Documentation

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

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

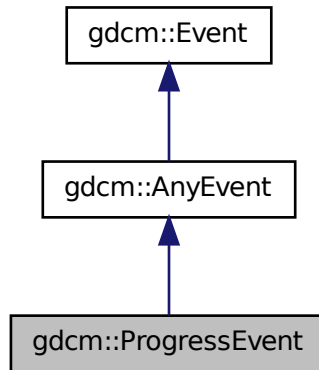
- gdcmPrivateTag.h

25.204 gdcm::ProgressEvent Class Reference

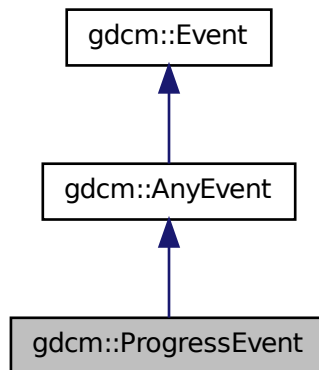
ProgressEvent Special type of event triggered during.

```
#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 ::gdcmm::Event *e) const
- virtual const char * GetEventName () const
- double GetProgress () const
- virtual ::gdcmm::Event * MakeObject () const
- void SetProgress (double p)

25.204.1 Detailed Description

ProgressEvent Special type of event triggered during.

See also

AnyEvent

25.204.2 Member Typedef Documentation

25.204.2.1 typedef ProgressEvent gdcmm::ProgressEvent::Self

25.204.2.2 typedef AnyEvent gdcmm::ProgressEvent::Superclass

25.204.3 Constructor & Destructor Documentation

25.204.3.1 gdcmm::ProgressEvent::ProgressEvent (double *p* = 0) [inline]

25.204.3.2 virtual gdcmm::ProgressEvent::~~ProgressEvent () [inline],[virtual]

25.204.3.3 gdcmm::ProgressEvent::ProgressEvent (const Self & *s*) [inline]

25.204.4 Member Function Documentation

25.204.4.1 virtual bool gdcmm::ProgressEvent::CheckEvent (const ::gdcmm::Event * *e*) const [inline],[virtual]

25.204.4.2 virtual const char* gdcmm::ProgressEvent::GetEventName () const [inline],[virtual]

Return the StringName associated with the event.

Implements gdcmm::Event.

25.204.4.3 double gdcmm::ProgressEvent::GetProgress () const [inline]

25.204.4.4 virtual ::gdcmm::Event* gdcmm::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 gdcmm::Event.

25.204.4.5 void gdcm::ProgressEvent::SetProgress (double *p*) [inline]

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

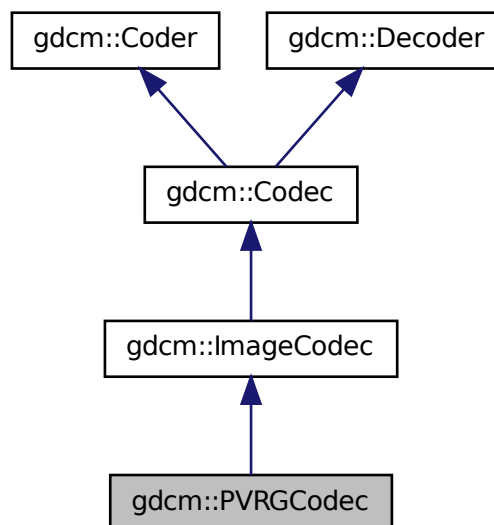
- gdcmProgressEvent.h

25.205 gdcm::PVRGCodec Class Reference

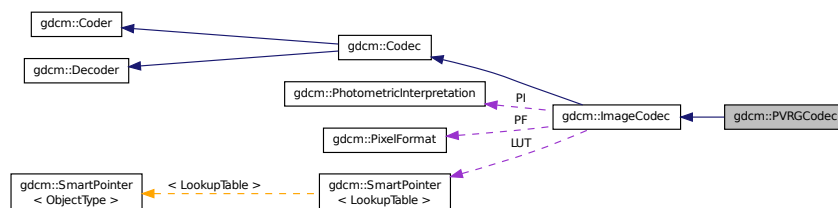
PVRGCodec.

```
#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)
- `bool Code (DataElement const &in, DataElement &out)`
Code.
- `bool Decode (DataElement const &is, DataElement &os)`
Decode.

Additional Inherited Members

25.205.1 Detailed Description

PVRGCodec.

Note

pvrp is a broken implementation of the JPEG standard. It is known to have a bug in the 16bits lossless implementation of the standard.

In an ideal world, you should not need this codec at all. But to support some broken file such as:

PHILIPS_Gyrosan-12-Jpeg_Extended_Process_2_4.dcm

we have to...

25.205.2 Constructor & Destructor Documentation

25.205.2.1 `gdcm::PVRGCodec::PVRGCodec ()`

25.205.2.2 `gdcm::PVRGCodec::~~PVRGCodec ()`

25.205.3 Member Function Documentation

25.205.3.1 `bool gdcm::PVRGCodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from `gdcm::ImageCodec`.

25.205.3.2 `bool gdcm::PVRGCodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from `gdcm::ImageCodec`.

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

Code.

Reimplemented from `gdcm::Coder`.

25.205.3.4 `bool gdcm::PVRGCodec::Decode (DataElement const & is_, DataElement & os) [virtual]`

Decode.

Reimplemented from `gdcm::ImageCodec`.

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

- `gdcmPVRGCodec.h`

25.206 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)`

25.206.1 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.

25.206.2 Constructor & Destructor Documentation

25.206.2.1 `gdcm::PythonFilter::PythonFilter ()`

25.206.2.2 `gdcm::PythonFilter::~~PythonFilter ()`

25.206.3 Member Function Documentation

25.206.3.1 `File& gdcm::PythonFilter::GetFile () [inline]`

25.206.3.2 `const File& gdcm::PythonFilter::GetFile () const` `[inline]`

25.206.3.3 `void gdcm::PythonFilter::SetDicts (const Dicts & dicts)`

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

25.206.3.5 `PyObject* gdcm::PythonFilter::ToPyObject (const Tag & t) const`

25.206.3.6 `void gdcm::PythonFilter::UseDictAlways (bool use)` `[inline]`

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

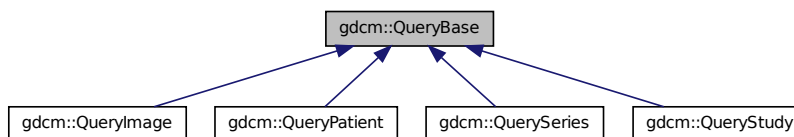
- `gdcmPythonFilter.h`

25.207 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 ()`
- `virtual std::vector< Tag > GetAllTags (const ERootType &inRootType) const`
- `virtual std::string 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`

25.207.1 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.

25.207.2 Constructor & Destructor Documentation

25.207.2.1 `virtual gdcmm::QueryBase::~QueryBase () [inline],[virtual]`

25.207.3 Member Function Documentation

25.207.3.1 `virtual std::vector<Tag> gdcmm::QueryBase::GetAllTags (const ERootType & inRootType) const [virtual]`

In order to validate a query dataset, just check for the presence of a tag, not it's requirement level in the spec

25.207.3.2 `virtual std::string gdcmm::QueryBase::GetName () const [pure virtual]`

Implemented in gdcmm::QueryImage, gdcmm::QueryPatient, gdcmm::QuerySeries, and gdcmm::QueryStudy.

25.207.3.3 `virtual std::vector<Tag> gdcmm::QueryBase::GetOptionalTags (const ERootType & inRootType) const [pure virtual]`

Implemented in gdcmm::QueryImage, gdcmm::QueryPatient, gdcmm::QuerySeries, and gdcmm::QueryStudy.

25.207.3.4 `virtual DataElement gdcmm::QueryBase::GetQueryLevel () const [pure virtual]`

Implemented in gdcmm::QueryImage, gdcmm::QueryPatient, gdcmm::QuerySeries, and gdcmm::QueryStudy.

25.207.3.5 `virtual std::vector<Tag> gdcmm::QueryBase::GetRequiredTags (const ERootType & inRootType) const [pure virtual]`

Implemented in gdcmm::QueryImage, gdcmm::QueryPatient, gdcmm::QuerySeries, and gdcmm::QueryStudy.

25.207.3.6 `virtual std::vector<Tag> gdcmm::QueryBase::GetUniqueTags (const ERootType & inRootType) const [pure virtual]`

Implemented in gdcmm::QueryImage, gdcmm::QueryPatient, gdcmm::QuerySeries, and gdcmm::QueryStudy.

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

- `gdcQueryBase.h`

25.208 gdc::QueryFactory Class Reference

`QueryFactory.h`.

```
#include <gdcQueryFactory.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 BaseRootQuery * ProduceQuery (ERootType inRootType, EQueryType inQueryType, EQueryLevel inQueryLevel)`

25.208.1 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

25.208.2 Member Function Documentation

25.208.2.1 `static ECharSet gdc::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()`).

25.208.2.2 `static void gdc::QueryFactory::ListCharSets (std::ostream & os) [static]`

List all possible `CharSet`.

25.208.2.3 `static DataElement gdc::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

25.208.2.4 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:

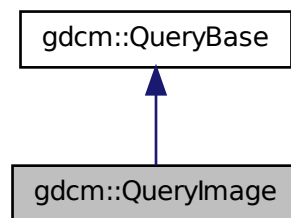
- gdcmQueryFactory.h

25.209 gdcm::QueryImage Class Reference

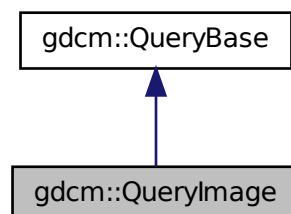
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

```
#include <gdcmQueryImage.h>
```

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for gdcm::QueryImage:



Public Member Functions

- `std::string 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`

25.209.1 Detailed Description

QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

25.209.2 Member Function Documentation

25.209.2.1 `std::string gdcM::QueryImage::GetName () const` `[inline],[virtual]`

Implements `gdcM::QueryBase`.

25.209.2.2 `std::vector<Tag> gdcM::QueryImage::GetOptionalTags (const ERootType & inRootType) const` `[virtual]`

Implements `gdcM::QueryBase`.

25.209.2.3 `DataElement gdcM::QueryImage::GetQueryLevel () const` `[virtual]`

Implements `gdcM::QueryBase`.

25.209.2.4 `std::vector<Tag> gdcM::QueryImage::GetRequiredTags (const ERootType & inRootType) const` `[virtual]`

Implements `gdcM::QueryBase`.

25.209.2.5 `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:

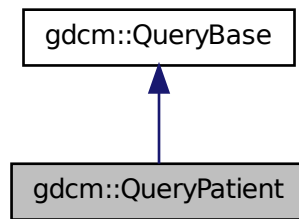
- `gdcMQueryImage.h`

25.210 gdcM::QueryPatient Class Reference

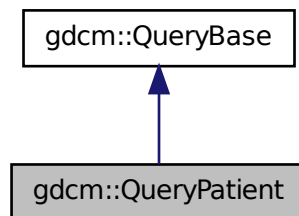
QueryPatient contains: class to construct a patient-based query for c-find and c-move.

```
#include <gdcMQueryPatient.h>
```

Inheritance diagram for gdcm::QueryPatient:



Collaboration diagram for gdcm::QueryPatient:



Public Member Functions

- `std::string 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`

25.210.1 Detailed Description

QueryPatient contains: class to construct a patient-based query for c-find and c-move.

25.210.2 Member Function Documentation

25.210.2.1 `std::string gdcm::QueryPatient::GetName () const` `[inline]`, `[virtual]`

Implements `gdcm::QueryBase`.

25.210.2.2 `std::vector<Tag> gdcmm::QueryPatient::GetOptionalTags (const ERootType & inRootType) const` [virtual]

Implements `gdcmm::QueryBase`.

25.210.2.3 `DataElement gdcmm::QueryPatient::GetQueryLevel () const` [virtual]

Implements `gdcmm::QueryBase`.

25.210.2.4 `std::vector<Tag> gdcmm::QueryPatient::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements `gdcmm::QueryBase`.

25.210.2.5 `std::vector<Tag> gdcmm::QueryPatient::GetUniqueTags (const ERootType & inRootType) const` [virtual]

Implements `gdcmm::QueryBase`.

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

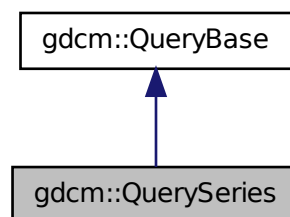
- `gdcmmQueryPatient.h`

25.211 gdcmm::QuerySeries Class Reference

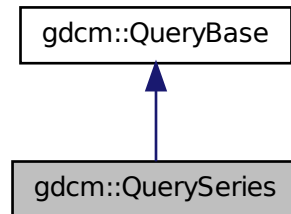
QuerySeries contains: class to construct a series-based query for c-find and c-move.

```
#include <gdcmmQuerySeries.h>
```

Inheritance diagram for `gdcmm::QuerySeries`:



Collaboration diagram for gdcm::QuerySeries:



Public Member Functions

- `std::string 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`

25.211.1 Detailed Description

QuerySeries contains: class to construct a series-based query for c-find and c-move.

25.211.2 Member Function Documentation

25.211.2.1 `std::string gdcm::QuerySeries::GetName () const` `[inline]`, `[virtual]`

Implements `gdcm::QueryBase`.

25.211.2.2 `std::vector<Tag> gdcm::QuerySeries::GetOptionalTags (const ERootType & inRootType) const` `[virtual]`

Implements `gdcm::QueryBase`.

25.211.2.3 `DataElement gdcm::QuerySeries::GetQueryLevel () const` `[virtual]`

Implements `gdcm::QueryBase`.

25.211.2.4 `std::vector<Tag> gdcm::QuerySeries::GetRequiredTags (const ERootType & inRootType) const` `[virtual]`

Implements `gdcm::QueryBase`.

25.211.2.5 `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:

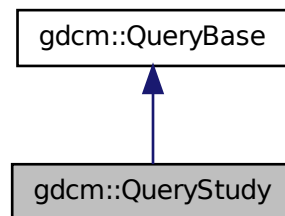
- `gdcmQuerySeries.h`

25.212 `gdcm::QueryStudy` Class Reference

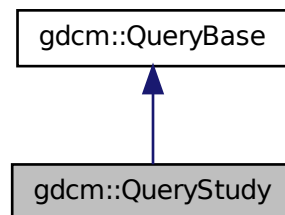
`QueryStudy.h` contains: class to construct a study-based query for C-FIND and C-MOVE.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for `gdcm::QueryStudy`:



Collaboration diagram for `gdcm::QueryStudy`:



Public Member Functions

- `std::string 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

25.212.1 Detailed Description

QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

25.212.2 Member Function Documentation

25.212.2.1 `std::string gdcm::QueryStudy::GetName () const` `[inline],[virtual]`

Implements gdcm::QueryBase.

25.212.2.2 `std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (const ERootType & inRootType) const` `[virtual]`

Implements gdcm::QueryBase.

25.212.2.3 `DataElement gdcm::QueryStudy::GetQueryLevel () const` `[virtual]`

Implements gdcm::QueryBase.

25.212.2.4 `std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (const ERootType & inRootType) const` `[virtual]`

Implements gdcm::QueryBase.

25.212.2.5 `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:

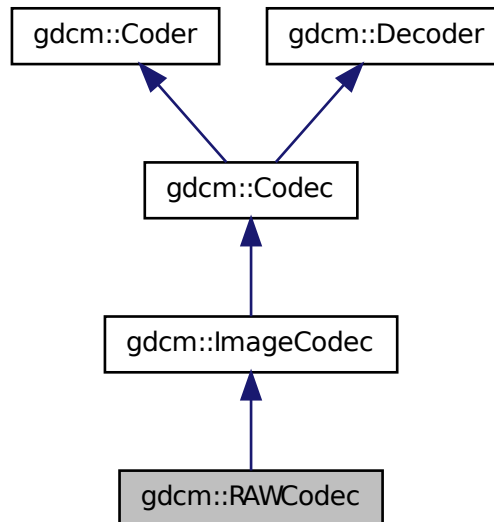
- gdcmQueryStudy.h

25.213 gdcm::RAWCodec Class Reference

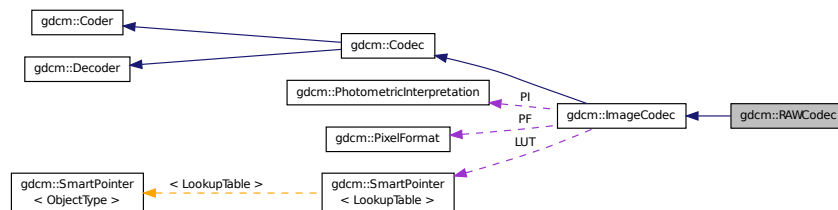
RAWCodec class.

```
#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)
- `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 Decode (std::istream &is, std::ostream &os)

Additional Inherited Members

25.213.1 Detailed Description

RAWCodec class.

25.213.2 Constructor & Destructor Documentation

25.213.2.1 gdcm::RAWCodec::RAWCodec ()

25.213.2.2 gdcm::RAWCodec::~~RAWCodec ()

25.213.3 Member Function Documentation

25.213.3.1 bool gdcm::RAWCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.213.3.2 bool gdcm::RAWCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

25.213.3.3 bool gdcm::RAWCodec::Code (DataElement const & in_, DataElement & out_) [virtual]

Code.

Reimplemented from gdcm::Coder.

25.213.3.4 bool gdcm::RAWCodec::Decode (DataElement const & is_, DataElement & os) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

25.213.3.5 bool gdcm::RAWCodec::Decode (std::istream & is, std::ostream & os) [protected],[virtual]

Reimplemented from gdcm::ImageCodec.

25.213.3.6 `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.

25.213.3.7 `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:

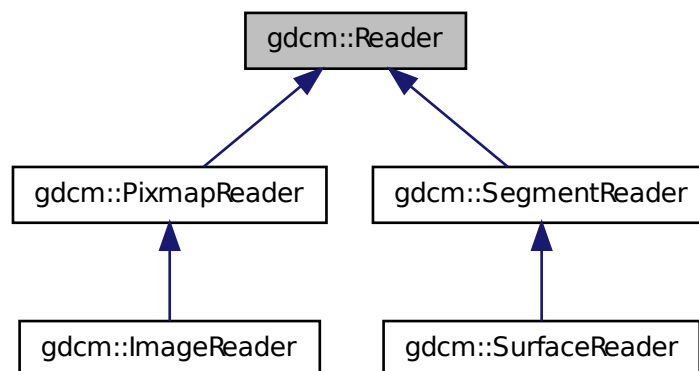
- `gdcmRAWCodec.h`

25.214 gdcm::Reader Class Reference

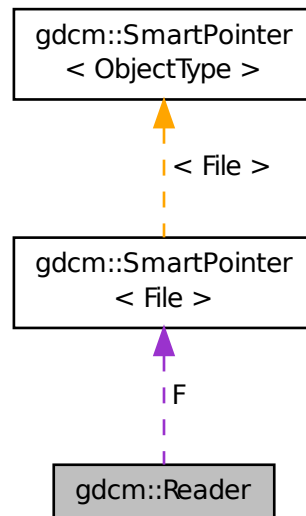
Reader ala DOM (Document Object Model)

```
#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.
- `virtual bool Read ()`
Main function to read a file.
- `bool ReadSelectedTags (std::set< Tag > const &tags)`
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

25.214.1 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:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2-K.cxx, gdcmrtonplan.cxx, gdcmrtpplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSD-O.cxx, ReadUTF8QtDir.cxx, rle2img.cxx, and TestReader.cxx.

25.214.2 Constructor & Destructor Documentation

25.214.2.1 `gdcm::Reader::Reader () [inline]`

25.214.2.2 `virtual gdcm::Reader::~Reader () [virtual]`

25.214.3 Member Function Documentation

25.214.3.1 `bool gdcm::Reader::CanRead () const`

Test whether this is a DICOM file

Warning

need to call either `SetFileName` or `SetStream` first

Examples:

`ReadUTF8QtDir.cxx`.

25.214.3.2 `const File& gdcm::Reader::GetFile () const [inline]`

Set/Get File.

Examples:

`ChangeSequenceUltrasound.cxx`, `ClinicalTrialAnnotate.cxx`, `CompressImage.cxx`, `csa2img.cxx`, `DiffFile.cxx`, `DumpADAC.cxx`, `DuplicatePCDE.cxx`, `ELSCINT1WaveToText.cxx`, `ExtractEncryptedContent.cxx`, `ExtractIconFromFile.cxx`, `FixBrokenJ2K.cxx`, `FixJAIBugJPEGLS.cxx`, `gdcmrtionplan.cxx`, `gdcmrtplan.cxx`, `GenLongSeqs.cxx`, `GenSeqs.cxx`, `GetJPEGSamplePrecision.cxx`, `GetSequenceUltrasound.cxx`, `HelloWorld.cxx`, `iU22tomultisc.cxx`, `LargeVRDS-Explicit.cxx`, `MergeTwoFiles.cxx`, `MrProtocol.cxx`, `PatchFile.cxx`, `pmsct_rgb1.cxx`, `ReadAndDumpDICOMDIR.cxx`, `ReadAndPrintAttributes.cxx`, `ReadExplicitLengthSQIVR.cxx`, `ReadGEMSSDO.cxx`, `rle2img.cxx`, and `TestReader.cxx`.

25.214.3.3 `File& gdcm::Reader::GetFile () [inline]`

Set/Get File.

25.214.3.4 `std::istream* gdcm::Reader::GetStreamPtr () const [inline],[protected]`

25.214.3.5 `virtual bool gdcm::Reader::Read () [virtual]`

Main function to read a file.

Reimplemented in `gdcm::PixmapReader`, `gdcm::ImageReader`, `gdcm::SegmentReader`, and `gdcm::SurfaceReader`.

Examples:

`ChangeSequenceUltrasound.cxx`, `ClinicalTrialAnnotate.cxx`, `csa2img.cxx`, `DiffFile.cxx`, `DumpADAC.cxx`, `DuplicatePCDE.cxx`, `ELSCINT1WaveToText.cxx`, `ExtractEncryptedContent.cxx`, `FixBrokenJ2K.cxx`, `gdcmrtionplan.cxx`, `gdcmrtplan.cxx`, `GenLongSeqs.cxx`, `GenSeqs.cxx`, `GetSequenceUltrasound.cxx`, `HelloWorld.cxx`, `iU22tomultisc.cxx`, `LargeVRDSEExplicit.cxx`, `PatchFile.cxx`, `pmsct_rgb1.cxx`, `ReadAndDumpDICOMDIR.cxx`, `ReadAndPrintAttributes.cxx`, `ReadExplicitLengthSQIVR.cxx`, `ReadGEMSSDO.cxx`, `rle2img.cxx`, and `TestReader.cxx`.

25.214.3.6 `bool gdcM::Reader::ReadDataSet ()` [protected]

25.214.3.7 `bool gdcM::Reader::ReadMetaInformation ()` [protected]

25.214.3.8 `bool gdcM::Reader::ReadPreamble ()` [protected]

25.214.3.9 `bool gdcM::Reader::ReadSelectedTags (std::set< Tag > const & tags)`

Will only read the specified selected tags.

25.214.3.10 `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>	

25.214.3.11 `void gdcM::Reader::SetFile (File & file)` [inline]

Set/Get File.

25.214.3.12 `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:

ChangeSequenceUltrasound.cxx, CheckBigEndianBug.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, ConvertToQImage.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DuplicatePC-DE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcMrtionplan.cxx, gdcMrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSample-Precision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, i-U22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadMultiTimesException.cxx, ReadUTF8QtDir.cxx, rle2img.cxx, TestReader.cxx, and threadgdcM.cxx.

25.214.3.13 `void gdcM::Reader::SetStream (std::istream & input_stream)` [inline]

Set the open-ed stream directly.

Examples:

ReadUTF8QtDir.cxx.

25.214.4 Friends And Related Function Documentation

25.214.4.1 friend class **StreamImageReader** [friend]

25.214.5 Member Data Documentation

25.214.5.1 **SmartPointer<File> gdcm::Reader::F** [protected]

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

- gdcmReader.h

25.215 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 ComputeInterceptSlopePixelType ()
- PixelFormat ComputePixelTypeFromMinMax ()
- 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 SetMinMaxForPixelType (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 SetTargetPixelType (PixelFormat const &targetst)
- void SetUseTargetPixelType (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)`

25.215.1 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-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

25.215.2 Constructor & Destructor Documentation

25.215.2.1 `gdcm::Rescaler::Rescaler ()` `[inline]`

25.215.2.2 `gdcm::Rescaler::~Rescaler ()` `[inline]`

25.215.3 Member Function Documentation

25.215.3.1 PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ()

Compute the Pixel Format of the output data Used for direct transformation

25.215.3.2 PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ()

Compute the Pixel Format of the output data Used for inverse transformation

25.215.3.3 double gdcm::Rescaler::GetIntercept () const [inline]**25.215.3.4 double gdcm::Rescaler::GetSlope () const [inline]****25.215.3.5 bool gdcm::Rescaler::InverseRescale (char * out, const char * in, size_t n)**

Inverse transform.

25.215.3.6 template<typename TIn > void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (char * out, const TIn * in, size_t n) [protected]**25.215.3.7 bool gdcm::Rescaler::Rescale (char * out, const char * in, size_t n)**

Direct transform.

25.215.3.8 template<typename TIn > void gdcm::Rescaler::RescaleFunctionIntoBestFit (char * out, const TIn * in, size_t n) [protected]**25.215.3.9 void gdcm::Rescaler::SetIntercept (double i) [inline]**

Set Intercept: used for both direct&inverse transformation.

25.215.3.10 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

25.215.3.11 void gdcm::Rescaler::SetPixelFormat (PixelFormat const & pf) [inline]

Set Pixel Format of input data.

25.215.3.12 void gdcm::Rescaler::SetSlope (double s) [inline]

Set Slope: user for both direct&inverse transformation.

25.215.3.13 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

25.215.3.14 void gdcM::Rescaler::SetUseTargetPixelType (bool b)

Override default behavior of Rescale.

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

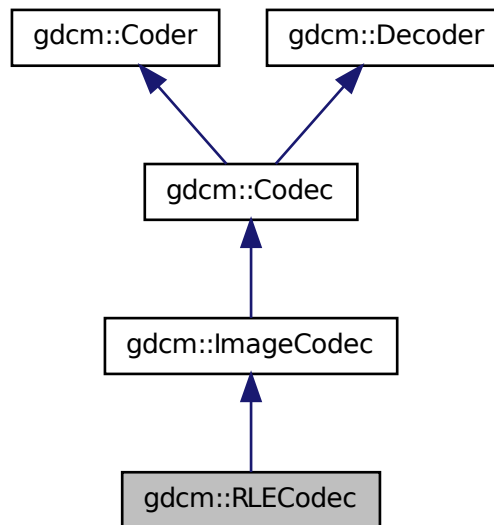
- gdcMRescaler.h

25.216 gdcM::RLECodec Class Reference

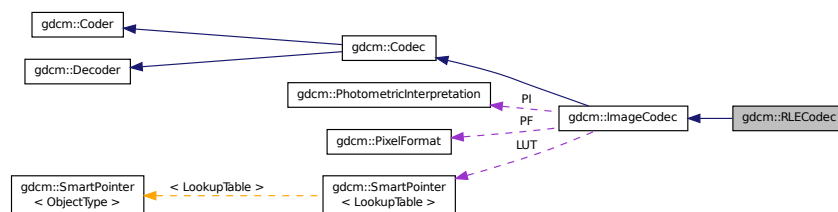
Class to do RLE.

```
#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)
- 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 Decode (std::istream &is, std::ostream &os)

Additional Inherited Members

25.216.1 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).

25.216.2 Constructor & Destructor Documentation

25.216.2.1 gdcm::RLECodec::RLECodec ()

25.216.2.2 gdcm::RLECodec::~~RLECodec ()

25.216.3 Member Function Documentation

25.216.3.1 bool gdcm::RLECodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

25.216.3.2 `bool gdcm::RLECodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from `gdcm::ImageCodec`.

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

Code.

Reimplemented from `gdcm::Coder`.

25.216.3.4 `bool gdcm::RLECodec::Decode (DataElement const & is_, DataElement & os)` [virtual]

Decode.

Reimplemented from `gdcm::ImageCodec`.

25.216.3.5 `bool gdcm::RLECodec::Decode (std::istream & is, std::ostream & os)` [protected],[virtual]

Reimplemented from `gdcm::ImageCodec`.

25.216.3.6 `unsigned long gdcm::RLECodec::GetBufferLength () const` [inline]

25.216.3.7 `bool gdcm::RLECodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from `gdcm::ImageCodec`.

25.216.3.8 `void gdcm::RLECodec::SetBufferLength (unsigned long /)` [inline]

25.216.3.9 `void gdcm::RLECodec::SetLength (unsigned long /)` [inline]

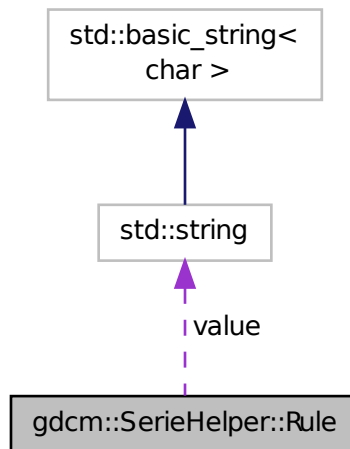
The documentation for this class was generated from the following file:

- `gdcmRLECodec.h`

25.217 gdcm::SerieHelper::Rule Struct Reference

```
#include <gdcmSerieHelper.h>
```

Collaboration diagram for gdcmm::SerieHelper::Rule:



Public Attributes

- `uint16_t` elem
- `uint16_t` group
- `int` op
- `std::string` value

25.217.1 Member Data Documentation

25.217.1.1 `uint16_t` gdcmm::SerieHelper::Rule::elem

25.217.1.2 `uint16_t` gdcmm::SerieHelper::Rule::group

25.217.1.3 `int` gdcmm::SerieHelper::Rule::op

25.217.1.4 `std::string` gdcmm::SerieHelper::Rule::value

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

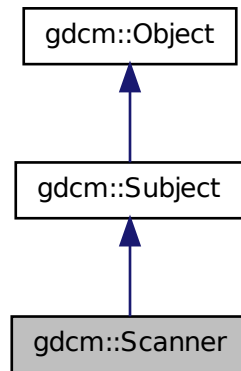
- `gdcmmSerieHelper.h`

25.218 gdcmm::Scanner Class Reference

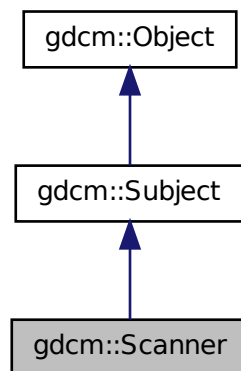
Scanner This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

```
#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 > ValueType

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 skip tags.*
- ConstIterator Begin () const
- void ClearSkipTags ()
- void ClearTags ()
- ConstIterator End () const
- Directory::FileNamesType GetAllFileNamesFromTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromTagToValue (Tag const &t, const char *valueref) const
- Directory::FileNamesType 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
- ValueType const & GetValues () const
 - Get all the values found (in lexicographic order)*
- ValueType 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)

25.218.1 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 gdcm::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, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

25.218.2 Member Typedef Documentation

25.218.2.1 `typedef MappingType::const_iterator gdcm::Scanner::ConstIterator`

25.218.2.2 `typedef std::map<const char *, TagToValue, Itstr> gdcm::Scanner::MappingType`

25.218.2.3 `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)

25.218.2.4 `typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType`

25.218.2.5 `typedef std::set< std::string > gdcm::Scanner::ValuesType`

25.218.3 Constructor & Destructor Documentation

25.218.3.1 `gdcm::Scanner::Scanner ()` `[inline]`

25.218.3.2 `gdcm::Scanner::~~Scanner ()`

25.218.4 Member Function Documentation

25.218.4.1 `void gdcm::Scanner::AddPrivateTag (PrivateTag const & t)`

25.218.4.2 `void gdcm::Scanner::AddSkipTag (Tag const & t)`

Add a tag that will need to be skipped. Those are root level skip tags.

25.218.4.3 `void gdcm::Scanner::AddTag (Tag const & t)`

Add a tag that will need to be read. Those are root level skip tags.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

25.218.4.4 `ConstIterator gdcm::Scanner::Begin () const` `[inline]`

25.218.4.5 `void gdcm::Scanner::ClearSkipTags ()`

25.218.4.6 `void gdcm::Scanner::ClearTags ()`

25.218.4.7 `ConstIterator gdcm::Scanner::End () const` `[inline]`

25.218.4.8 `Directory::FileNamesType gdcm::Scanner::GetAllFileNamesFromTagToValue (Tag const & t, const char * valuref) const`

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valuref'

25.218.4.9 `const char* gdcm::Scanner::GetFilenameFromTagToValue (Tag const & t, const char * valuref) const`

Will loop over all files and return the first file where value match the reference value 'valuref'

25.218.4.10 `Directory::FileNamesType const& gdcm::Scanner::GetFileNames () const` `[inline]`

25.218.4.11 `Directory::FileNamesType gdcm::Scanner::GetKeys () const`

Return the list of filename that are key in the internal map, which means those filename were properly parsed

Examples:

VolumeSorter.cxx.

25.218.4.12 TagToValue const& gdcm::Scanner::GetMapping (const char * *filename*) const

Get the std::map mapping filenames to value for file 'filename'.

Examples:

DumpToSQLITE3.cxx, and SimpleScanner.cxx.

25.218.4.13 TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (Tag const & *t*, const char * *value*) const

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

25.218.4.14 MappingType const& gdcm::Scanner::GetMappings () const [inline]

Mappings are the mapping from a particular tag to the map, mapping filename to value:

25.218.4.15 Directory::FileNamesType gdcm::Scanner::GetOrderedValues (Tag const & *t*) const

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

25.218.4.16 const char* gdcm::Scanner::GetValue (const char * *filename*, Tag const & *t*) const

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

25.218.4.17 ValuesType const& gdcm::Scanner::GetValues () const [inline]

Get all the values found (in lexicographic order)

Examples:

SortImage.cxx, and VolumeSorter.cxx.

25.218.4.18 ValuesType gdcm::Scanner::GetValues (Tag const & *t*) const

Get all the values found (in lexicographic order) associated with Tag 't'.

25.218.4.19 `bool gdcm::Scanner::IsKey (const char * filename) const`

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

Examples:

DumpToSQLITE3.cxx, and SimpleScanner.cxx.

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

for wrapped language: instantiate a reference counted object

25.218.4.21 `void gdcm::Scanner::Print (std::ostream & os) const [virtual]`

Print result.

Reimplemented from `gdcm::Object`.

Referenced by `gdcm::operator<<()`.

25.218.4.22 `void gdcm::Scanner::ProcessPublicTag (StringFilter & sf, const char * filename) [protected]`

25.218.4.23 `bool gdcm::Scanner::Scan (Directory::FileNamesType const & filenames)`

Start the scan !

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

25.218.5 Friends And Related Function Documentation

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

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

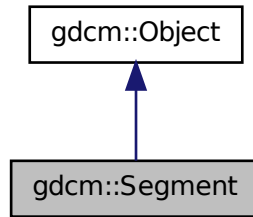
- `gdcmScanner.h`

25.219 gdcm::Segment Class Reference

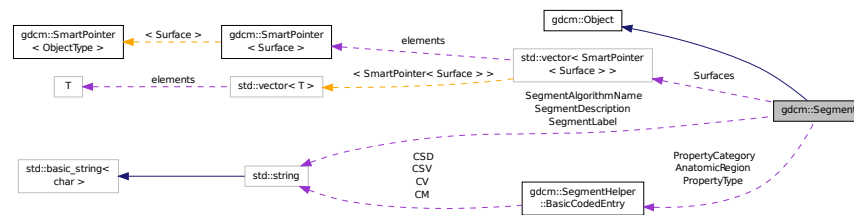
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

```
#include <gdcmSegment.h>
```

Inheritance diagram for `gdcM::Segment`:



Collaboration diagram for `gdcM::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

25.219.1 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

25.219.2 Member Typedef Documentation

25.219.2.1 `typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector`

25.219.3 Member Enumeration Documentation

25.219.3.1 `enum gdcm::Segment::ALGOType`

Enumerator:

MANUAL
AUTOMATIC
ALGOType_END

25.219.4 Constructor & Destructor Documentation

25.219.4.1 `gdcm::Segment::Segment ()`

25.219.4.2 `virtual gdcm::Segment::~~Segment () [virtual]`

25.219.5 Member Function Documentation

25.219.5.1 `void gdcm::Segment::AddSurface (SmartPointer< Surface > surface)`

25.219.5.2 `static ALGOType gdcm::Segment::GetALGOType (const char * type) [static]`

25.219.5.3 `static const char* gdcm::Segment::GetALGOTypeString (ALGOType type) [static]`

25.219.5.4 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion () const`

25.219.5.5 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ()`

25.219.5.6 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory () const`

25.219.5.7 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ()`

25.219.5.8 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType () const`

25.219.5.9 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ()`

25.219.5.10 `const char* gdcm::Segment::GetSegmentAlgorithmName () const`

25.219.5.11 `ALGOType gdcm::Segment::GetSegmentAlgorithmType () const`

25.219.5.12 `const char* gdcm::Segment::GetSegmentDescription () const`

25.219.5.13 `const char* gdcm::Segment::GetSegmentLabel () const`

25.219.5.14 `unsigned short gdcm::Segment::GetSegmentNumber () const`

25.219.5.15 `SmartPointer< Surface > gdcm::Segment::GetSurface (const unsigned int idx = 0) const`

- 25.219.5.16 unsigned long gdcm::Segment::GetSurfaceCount ()
- 25.219.5.17 SurfaceVector const& gdcm::Segment::GetSurfaces () const
- 25.219.5.18 SurfaceVector& gdcm::Segment::GetSurfaces ()
- 25.219.5.19 void gdcm::Segment::SetAnatomicRegion (SegmentHelper::BasicCodedEntry const & *BSE*)
- 25.219.5.20 void gdcm::Segment::SetPropertyCategory (SegmentHelper::BasicCodedEntry const & *BSE*)
- 25.219.5.21 void gdcm::Segment::SetPropertyType (SegmentHelper::BasicCodedEntry const & *BSE*)
- 25.219.5.22 void gdcm::Segment::SetSegmentAlgorithmName (const char * *name*)
- 25.219.5.23 void gdcm::Segment::SetSegmentAlgorithmType (ALGOType *type*)
- 25.219.5.24 void gdcm::Segment::SetSegmentAlgorithmType (const char * *typeStr*)
- 25.219.5.25 void gdcm::Segment::SetSegmentDescription (const char * *description*)
- 25.219.5.26 void gdcm::Segment::SetSegmentLabel (const char * *label*)
- 25.219.5.27 void gdcm::Segment::SetSegmentNumber (const unsigned short *num*)
- 25.219.5.28 void gdcm::Segment::SetSurfaceCount (const unsigned long *nb*)

25.219.6 Member Data Documentation

- 25.219.6.1 SegmentHelper::BasicCodedEntry gdcm::Segment::AnatomicRegion [protected]
- 25.219.6.2 SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyCategory [protected]
- 25.219.6.3 SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyType [protected]
- 25.219.6.4 std::string gdcm::Segment::SegmentAlgorithmName [protected]
- 25.219.6.5 ALGOType gdcm::Segment::SegmentAlgorithmType [protected]
- 25.219.6.6 std::string gdcm::Segment::SegmentDescription [protected]
- 25.219.6.7 std::string gdcm::Segment::SegmentLabel [protected]
- 25.219.6.8 unsigned short gdcm::Segment::SegmentNumber [protected]
- 25.219.6.9 unsigned long gdcm::Segment::SurfaceCount [protected]
- 25.219.6.10 SurfaceVector gdcm::Segment::Surfaces [protected]

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

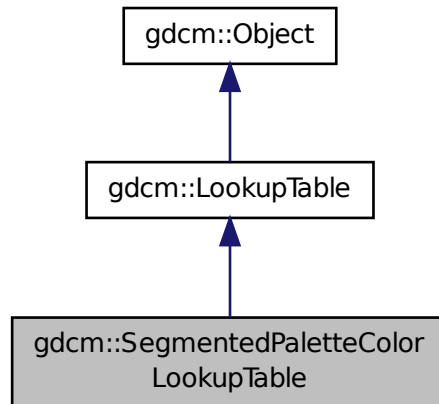
- gdcmSegment.h

25.220 gdcM::SegmentedPaletteColorLookupTable Class Reference

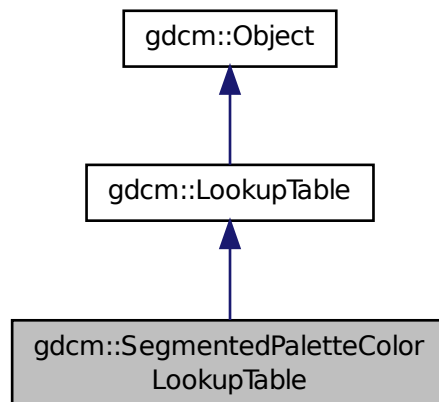
SegmentedPaletteColorLookupTable class.

```
#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

25.220.1 Detailed Description

SegmentedPaletteColorLookupTable class.

25.220.2 Constructor & Destructor Documentation

25.220.2.1 gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ()

25.220.2.2 gdcm::SegmentedPaletteColorLookupTable::~~SegmentedPaletteColorLookupTable ()

25.220.3 Member Function Documentation

25.220.3.1 void gdcm::SegmentedPaletteColorLookupTable::Print (std::ostream &) const [inline],[virtual]

Reimplemented from gdcm::LookupTable.

25.220.3.2 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:

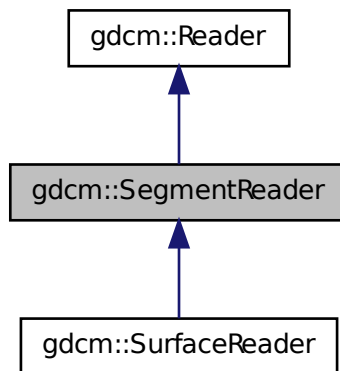
- gdcmSegmentedPaletteColorLookupTable.h

25.221 gdcm::SegmentReader Class Reference

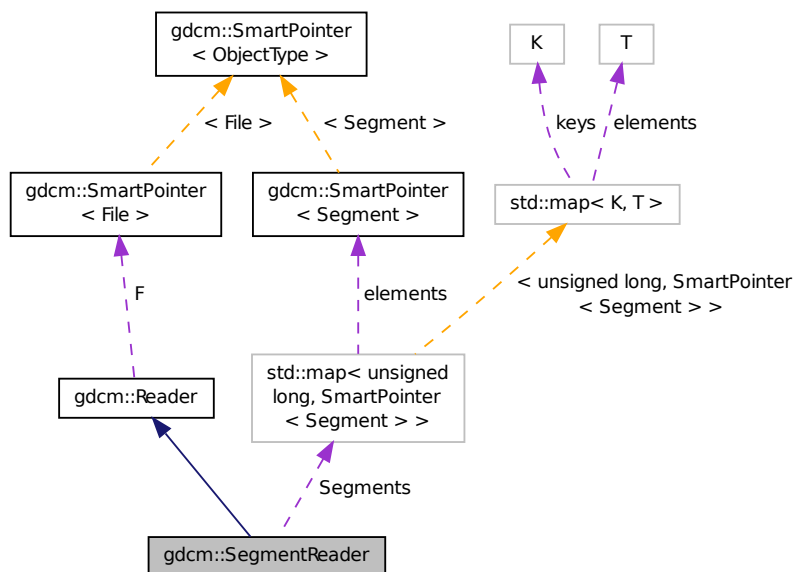
This class defines a segment reader. It reads attributes of group 0x0062.

```
#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`

25.221.1 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

25.221.2 Member Typedef Documentation

25.221.2.1 `typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap`
[protected]

25.221.2.2 `typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector`

25.221.3 Constructor & Destructor Documentation

25.221.3.1 `gdcm::SegmentReader::SegmentReader ()`

25.221.3.2 `virtual gdcm::SegmentReader::~~SegmentReader ()` [virtual]

25.221.4 Member Function Documentation

25.221.4.1 `const SegmentVector gdcm::SegmentReader::GetSegments () const`

25.221.4.2 **SegmentVector** gdcmm::SegmentReader::GetSegments ()

25.221.4.3 **virtual bool** gdcmm::SegmentReader::Read () [virtual]

Read.

Reimplemented from gdcmm::Reader.

Reimplemented in gdcmm::SurfaceReader.

25.221.4.4 **bool** gdcmm::SegmentReader::ReadSegment (const Item & *segmentItem*, const unsigned int *idx*) [protected]

25.221.4.5 **bool** gdcmm::SegmentReader::ReadSegments () [protected]

25.221.5 Member Data Documentation

25.221.5.1 **SegmentMap** gdcmm::SegmentReader::Segments [protected]

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

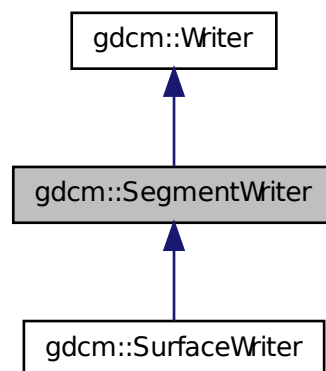
- gdcmmSegmentReader.h

25.222 gdcmm::SegmentWriter Class Reference

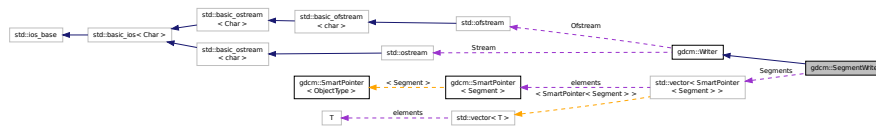
This class defines a segment writer. It writes attributes of group 0x0062.

```
#include <gdcmmSegmentWriter.h>
```

Inheritance diagram for gdcmm::SegmentWriter:



Collaboration diagram for gdcm::SegmentWriter:



Public Types

- typedef std::vector
 < SmartPointer< Segment > > SegmentVector

Public Member Functions

- SegmentWriter ()
- virtual ~SegmentWriter ()
- void AddSegment (SmartPointer< Segment > segment)
- unsigned int GetNumberOfSegments () const
- SmartPointer< Segment > GetSegment (const unsigned int idx=0) const
- const SegmentVector & GetSegments () const
- SegmentVector & GetSegments ()
- void SetNumberOfSegments (const unsigned int size)
- void SetSegments (SegmentVector &segments)
- bool Write ()

Write.

Protected Member Functions

- bool PrepareWrite ()

Protected Attributes

- SegmentVector Segments

25.222.1 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

25.222.2 Member Typedef Documentation

25.222.2.1 `typedef std::vector< SmartPointer< Segment > > gdcM::SegmentWriter::SegmentVector`

25.222.3 Constructor & Destructor Documentation

25.222.3.1 `gdcM::SegmentWriter::SegmentWriter ()`

25.222.3.2 `virtual gdcM::SegmentWriter::~~SegmentWriter () [virtual]`

25.222.4 Member Function Documentation

25.222.4.1 `void gdcM::SegmentWriter::AddSegment (SmartPointer< Segment > segment)`

25.222.4.2 `unsigned int gdcM::SegmentWriter::GetNumberOfSegments () const`

25.222.4.3 `SmartPointer< Segment > gdcM::SegmentWriter::GetSegment (const unsigned int idx = 0) const`

25.222.4.4 `const SegmentVector& gdcM::SegmentWriter::GetSegments () const`

25.222.4.5 `SegmentVector& gdcM::SegmentWriter::GetSegments ()`

25.222.4.6 `bool gdcM::SegmentWriter::PrepareWrite () [protected]`

Reimplemented in `gdcM::SurfaceWriter`.

25.222.4.7 `void gdcM::SegmentWriter::SetNumberOfSegments (const unsigned int size)`

25.222.4.8 `void gdcM::SegmentWriter::SetSegments (SegmentVector & segments)`

25.222.4.9 `bool gdcM::SegmentWriter::Write () [virtual]`

Write.

Reimplemented from `gdcM::Writer`.

Reimplemented in `gdcM::SurfaceWriter`.

25.222.5 Member Data Documentation

25.222.5.1 `SegmentVector gdcM::SegmentWriter::Segments [protected]`

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

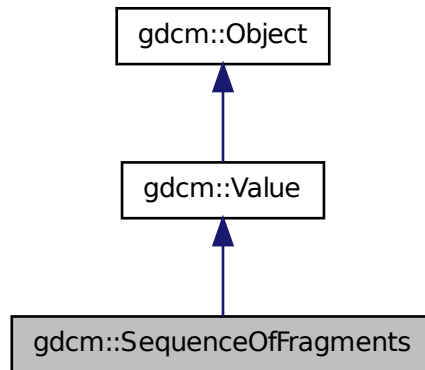
- `gdcMSegmentWriter.h`

25.223 gdcM::SequenceOfFragments Class Reference

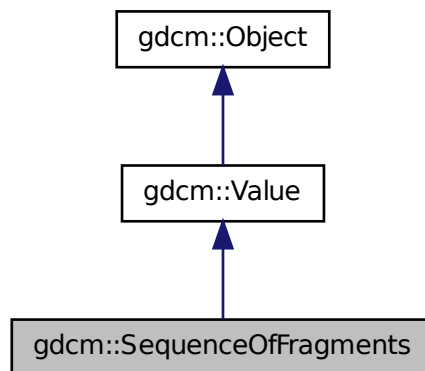
Class to represent a Sequence Of Fragments.

```
#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)
- 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 ()

25.223.1 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.

25.223.2 Member Typedef Documentation

25.223.2.1 `typedef FragmentVector::const_iterator gdcmm::SequenceOfFragments::ConstIterator`

25.223.2.2 `typedef std::vector<Fragment> gdcmm::SequenceOfFragments::FragmentVector`

25.223.2.3 `typedef FragmentVector::iterator gdcmm::SequenceOfFragments::Iterator`

25.223.2.4 `typedef FragmentVector::size_type gdcmm::SequenceOfFragments::SizeType`

25.223.3 Constructor & Destructor Documentation

25.223.3.1 `gdcmm::SequenceOfFragments::SequenceOfFragments () [inline]`

constructor (UndefinedLength by default)

25.223.4 Member Function Documentation

25.223.4.1 `void gdcmm::SequenceOfFragments::AddFragment (Fragment const & item)`

Appends a Fragment to the already added ones.

Examples:

FixBrokenJ2K.cxx.

25.223.4.2 `Iterator gdcmm::SequenceOfFragments::Begin () [inline]`

25.223.4.3 `ConstIterator gdcmm::SequenceOfFragments::Begin () const [inline]`

25.223.4.4 `void gdcmm::SequenceOfFragments::Clear () [virtual]`

Clear.

Implements gdcmm::Value.

25.223.4.5 `unsigned long gdcmm::SequenceOfFragments::ComputeByteLength () const`

25.223.4.6 `VL gdcmm::SequenceOfFragments::ComputeLength () const`

25.223.4.7 `Iterator gdcmm::SequenceOfFragments::End () [inline]`

25.223.4.8 `ConstIterator gdcmm::SequenceOfFragments::End () const [inline]`

25.223.4.9 `bool gdcmm::SequenceOfFragments::GetBuffer (char * buffer, unsigned long length) const`

25.223.4.10 `bool gdcmm::SequenceOfFragments::GetFragBuffer (unsigned int fragNb, char * buffer, unsigned long & length) const`

25.223.4.11 `const Fragment& gdcM::SequenceOfFragments::GetFragment (SizeType num) const`

Examples:

FixBrokenJ2K.cxx, and FixJAIBugJPEGs.cxx.

25.223.4.12 `VL gdcM::SequenceOfFragments::GetLength () const [inline],[virtual]`

Returns the SQ length, as read from disk.

Implements gdcM::Value.

25.223.4.13 `SizeType gdcM::SequenceOfFragments::GetNumberOfFragments () const`

Examples:

FixJAIBugJPEGs.cxx.

25.223.4.14 `const BasicOffsetTable& gdcM::SequenceOfFragments::GetTable () const [inline]`

25.223.4.15 `BasicOffsetTable& gdcM::SequenceOfFragments::GetTable () [inline]`

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

25.223.4.17 `bool gdcM::SequenceOfFragments::operator== (const Value & val) const [inline],[virtual]`

Implements gdcM::Value.

25.223.4.18 `void gdcM::SequenceOfFragments::Print (std::ostream & os) const [inline],[virtual]`

Reimplemented from gdcM::Object.

25.223.4.19 `template<typename TSwap > std::istream& gdcM::SequenceOfFragments::Read (std::istream & is) [inline]`

References gdcMDebugMacro, gdcMWarningMacro, gdcM::DataElement::GetTag(), gdcM::DataElement::GetVL(), gdcM::Fragment::Read(), gdcM::DataElement::SetByteValue(), and gdcM::Exception::what().

25.223.4.20 `void gdcM::SequenceOfFragments::SetLength (VL length) [inline],[virtual]`

Sets the actual SQ length.

Implements gdcM::Value.

25.223.4.21 `template<typename TSwap > std::ostream const& gdcM::SequenceOfFragments::Write (std::ostream & os) const [inline]`

References gdcM::VL::Write(), and gdcM::Tag::Write().

25.223.4.22 `bool gdcm::SequenceOfFragments::WriteBuffer (std::ostream & os) const`

Examples:

GetJPEGSamplePrecision.cxx.

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

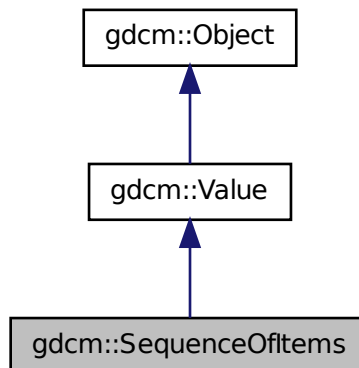
- gdcmSequenceOfFragments.h

25.224 gdcm::SequenceOfItems Class Reference

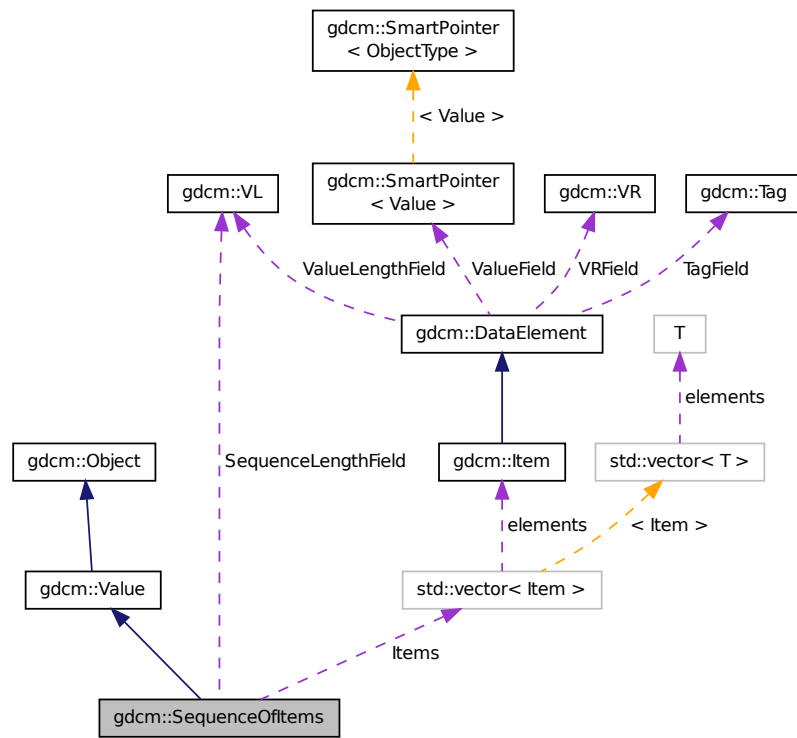
Class to represent a Sequence Of Items (value representation : SQ)

```
#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.
- `Iterator Begin ()`
- `ConstIterator Begin () const`
- `void Clear ()`
- `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)
- 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.

25.224.1 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:

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.

25.224.2 Member Typedef Documentation

25.224.2.1 `typedef ItemVector::const_iterator gdcmm::SequenceOfItems::ConstIterator`

25.224.2.2 `typedef std::vector< Item > gdcmm::SequenceOfItems::ItemVector`

25.224.2.3 `typedef ItemVector::iterator gdcmm::SequenceOfItems::Iterator`

25.224.2.4 `typedef ItemVector::size_type gdcmm::SequenceOfItems::SizeType`

25.224.3 Constructor & Destructor Documentation

25.224.3.1 `gdcmm::SequenceOfItems::SequenceOfItems () [inline]`

constructor (UndefinedLength by default)

25.224.4 Member Function Documentation

25.224.4.1 `void gdcmm::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.

25.224.4.2 `Iterator gdcmm::SequenceOfItems::Begin () [inline]`

25.224.4.3 `ConstIterator gdcmm::SequenceOfItems::Begin () const [inline]`

25.224.4.4 `void gdcmm::SequenceOfItems::Clear () [inline],[virtual]`

Implements `gdcmm::Value`.

25.224.4.5 `template<typename TDE > VL gdcmm::SequenceOfItems::ComputeLength () const`

25.224.4.6 `Iterator gdcmm::SequenceOfItems::End () [inline]`

25.224.4.7 `ConstIterator gdcmm::SequenceOfItems::End () const [inline]`

25.224.4.8 `bool gdcmm::SequenceOfItems::FindDataElement (const Tag & t) const`

25.224.4.9 `const Item& gdcmm::SequenceOfItems::GetItem (SizeType position) const`

Examples:

ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, gdcmmrtionplan.cxx, gdcmmrtplan.cxx, GetSequenceUltrasound.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

25.224.4.10 `Item& gdcmm::SequenceOfItems::GetItem (SizeType position)`

25.224.4.11 `VL gdcmm::SequenceOfItems::GetLength () const [inline],[virtual]`

Returns the SQ length, as read from disk.

Implements `gdcmm::Value`.

25.224.4.12 `SizeType gdcmm::SequenceOfItems::GetNumberOfItems () const [inline]`

Examples:

`ChangeSequenceUltrasound.cxx`, `DumpGEMSMovieGroup.cxx`, `ExtractEncryptedContent.cxx`, `gdcmmrtionplan.cxx`, `gdcmmrtplan.cxx`, `GetSequenceUltrasound.cxx`, and `LargeVRDSExplicit.cxx`.

25.224.4.13 `bool gdcmm::SequenceOfItems::IsUndefinedLength () const [inline]`

return if Value Length if of undefined length

25.224.4.14 `static SmartPointer<SequenceOfItems> gdcmm::SequenceOfItems::New () [inline],[static]`

25.224.4.15 `SequenceOfItems& gdcmm::SequenceOfItems::operator= (const SequenceOfItems & val) [inline]`

References `Items`, and `SequenceLengthField`.

25.224.4.16 `bool gdcmm::SequenceOfItems::operator== (const Value & val) const [inline],[virtual]`

Implements `gdcmm::Value`.

References `Items`, and `SequenceLengthField`.

25.224.4.17 `void gdcmm::SequenceOfItems::Print (std::ostream & os) const [inline],[virtual]`

Reimplemented from `gdcmm::Object`.

25.224.4.18 `template<typename TDE , typename TSwap > std::istream& gdcmm::SequenceOfItems::Read (std::istream & is) [inline]`

Examples:

`ReadExplicitLengthSQIVR.cxx`.

References `gdcmm::Item::Clear()`, `gdcmmDebugMacro`, `gdcmmWarningMacro`, `gdcmm::Exception::GetDescription()`, `gdcmm::Item::GetNestedDataSet()`, `gdcmm::DataElement::GetTag()`, `gdcmm::DataElement::GetVL()`, `gdcmm::Item::Read()`, and `gdcmm::DataSet::Size()`.

25.224.4.19 `void gdcmm::SequenceOfItems::SetLength (VL length) [inline],[virtual]`

Sets the actual SQ length.

Implements `gdcmm::Value`.

Examples:

ReadExplicitLengthSQIVR.cxx.

25.224.4.20 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, GenAllVR.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

25.224.4.21 void gdcm::SequenceOfItems::SetNumberOfItems (SizeType *n*) [inline]

25.224.4.22 template<typename TDE , typename TSwap > std::ostream const& gdcm::SequenceOfItems::Write (std::ostream & *os*) const [inline]

References gdcm::VL::Write(), and gdcm::Tag::Write().

25.224.5 Member Data Documentation

25.224.5.1 ItemVector gdcm::SequenceOfItems::Items

Vector of Sequence Items.

Referenced by operator=(), and operator==().

25.224.5.2 VL gdcm::SequenceOfItems::SequenceLengthField

Total length of the Sequence (or 0xffffffff if undefined).

Referenced by operator=(), and operator==().

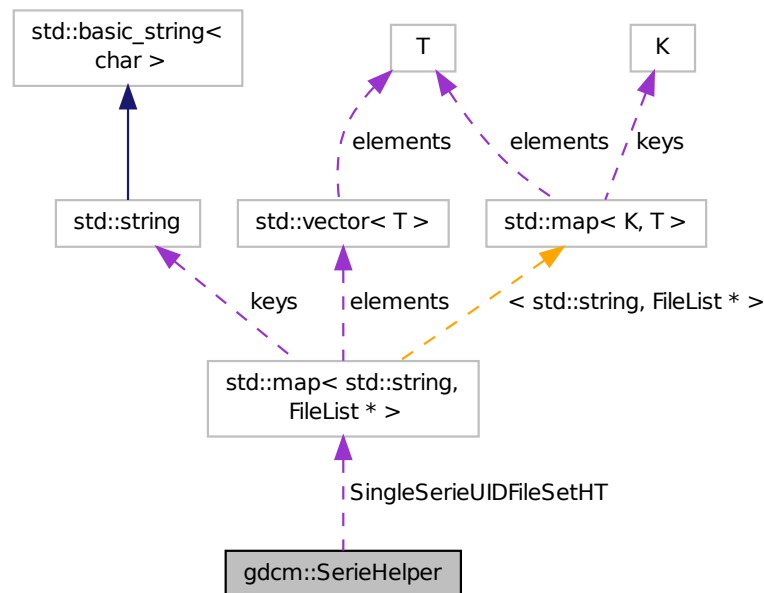
The documentation for this class was generated from the following file:

- gdcmSequenceOfItems.h

25.225 gdcm::SerieHelper Class Reference

```
#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`

25.225.1 Detailed Description

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 `gdcm::ImageHelper` or `gdcm::IPPSorter`

25.225.2 Member Typedef Documentation

25.225.2.1 `typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions` [protected]

25.225.2.2 `typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap` [protected]

25.225.3 Constructor & Destructor Documentation

25.225.3.1 `gdcm::SerieHelper::SerieHelper ()`

25.225.3.2 `gdcm::SerieHelper::~~SerieHelper ()`

25.225.4 Member Function Documentation

25.225.4.1 `bool gdcm::SerieHelper::AddFile (FileWithName & header)` [protected]

25.225.4.2 `void gdcm::SerieHelper::AddFileName (std::string const & filename)` [protected]

25.225.4.3 `void gdcm::SerieHelper::AddRestriction (const std::string & tag)`

25.225.4.4 `void gdcm::SerieHelper::AddRestriction (uint16_t group, uint16_t elem, std::string const & value, int op)`

- 25.225.4.5 void gdcm::SerieHelper::AddRestriction (const Tag & tag) [protected]
- 25.225.4.6 void gdcm::SerieHelper::Clear ()
- 25.225.4.7 void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ()
- 25.225.4.8 std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (File * inFile)
- 25.225.4.9 bool gdcm::SerieHelper::FileNameOrdering (FileList * fileList) [protected]
- 25.225.4.10 FileList* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ()
- 25.225.4.11 FileList* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ()
- 25.225.4.12 bool gdcm::SerieHelper::ImagePositionPatientOrdering (FileList * fileSet) [protected]
- 25.225.4.13 void gdcm::SerieHelper::OrderFileList (FileList * fileSet)
- 25.225.4.14 void gdcm::SerieHelper::SetDirectory (std::string const & dir, bool recursive = false)
- 25.225.4.15 void gdcm::SerieHelper::SetLoadMode (int) [inline]
- 25.225.4.16 void gdcm::SerieHelper::SetUseSeriesDetails (bool useSeriesDetails)
- 25.225.4.17 bool gdcm::SerieHelper::UserOrdering (FileList * fileSet) [protected]

25.225.5 Member Data Documentation

- 25.225.5.1 SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt [protected]
- 25.225.5.2 SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT [protected]

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

- gdcmSerieHelper.h

25.226 gdcm::Series Class Reference

Series.

```
#include <gdcmSeries.h>
```

Public Member Functions

- Series ()

25.226.1 Detailed Description

Series.

25.226.2 Constructor & Destructor Documentation

25.226.2.1 `gdcm::Series::Series ()` `[inline]`

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

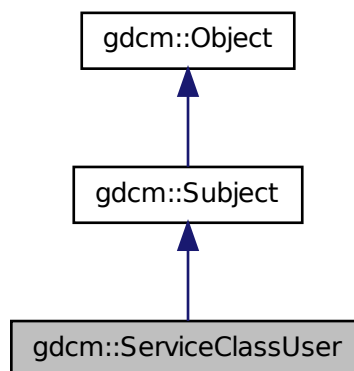
- `gdcmSeries.h`

25.227 `gdcm::ServiceClassUser` Class Reference

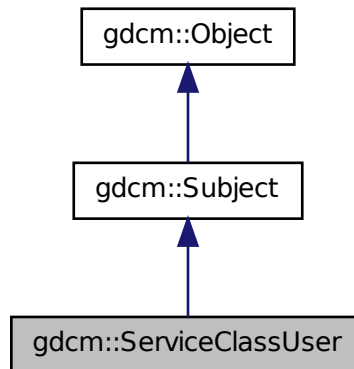
`ServiceClassUser`.

```
#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
- time_t GetTimeout () const
- bool InitializeConnection ()
- 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 (time_t t)
 - set/get Timeout*
 - bool StartAssociation ()
 - Start the association. Need to call SetPresentationContexts before.*
 - bool StopAssociation ()
 - Stop the running association.*

25.227.1 Detailed Description

ServiceClassUser.

Examples:

CStoreQtProgress.cxx.

25.227.2 Constructor & Destructor Documentation

25.227.2.1 gdcm::ServiceClassUser::ServiceClassUser ()

Construct a SCU with default:

- hostname = localhost
- port = 104

25.227.2.2 gdcm::ServiceClassUser::~~ServiceClassUser ()

25.227.3 Member Function Documentation

25.227.3.1 const char* gdcm::ServiceClassUser::GetAETitle () const

25.227.3.2 const char* gdcm::ServiceClassUser::GetCalledAETitle () const

25.227.3.3 time_t gdcm::ServiceClassUser::GetTimeout () const

25.227.3.4 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.

25.227.3.5 `bool gdcm::ServiceClassUser::SendEcho ()`

C-ECHO.

25.227.3.6 `bool gdcm::ServiceClassUser::SendFind (const BaseRootQuery * query, std::vector< DataSet > & retDatasets)`

C-FIND a query, return result are in retDatasets.

25.227.3.7 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, const char * outputdir)`

Execute a C-MOVE, based on query, return files are written in outputdir.

25.227.3.8 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, std::vector< DataSet > & retDatasets)`

Execute a C-MOVE, based on query, returned dataset are Implicit.

25.227.3.9 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, std::vector< File > & retFile)`

Execute a C-MOVE, based on query, returned Files are stored in vector.

25.227.3.10 `bool gdcm::ServiceClassUser::SendStore (const char * filename)`

Execute a C-STORE on file on disk, named filename.

Examples:

CStoreQtProgress.cxx.

25.227.3.11 `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.

25.227.3.12 `bool gdcm::ServiceClassUser::SendStore (DataSet const & ds)`

Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.

25.227.3.13 `void gdcm::ServiceClassUser::SetAETitle (const char * aetitle)`

set calling ae title

25.227.3.14 `void gdcm::ServiceClassUser::SetCalledAETitle (const char * aetitle)`

set called ae title

Examples:

CStoreQtProgress.cxx.

25.227.3.15 void gdcm::ServiceClassUser::SetHostname (const char * *hostname*)

Set the name of the called hostname (hostname or IP address)

Examples:

CStoreQtProgress.cxx.

25.227.3.16 void gdcm::ServiceClassUser::SetPort (uint16_t *port*)

Set port of remote host (called application)

Examples:

CStoreQtProgress.cxx.

25.227.3.17 void gdcm::ServiceClassUser::SetPortSCP (uint16_t *portscp*)

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

25.227.3.18 void gdcm::ServiceClassUser::SetPresentationContexts (std::vector< PresentationContext > const & *pcs*)

Set the Presentation Context used for the Association.

Examples:

CStoreQtProgress.cxx.

25.227.3.19 void gdcm::ServiceClassUser::SetTimeout (time_t *t*)

set/get Timeout

Examples:

CStoreQtProgress.cxx.

25.227.3.20 bool gdcm::ServiceClassUser::StartAssociation ()

Start the association. Need to call SetPresentationContexts before.

Examples:

CStoreQtProgress.cxx.

25.227.3.21 `bool gdcm::ServiceClassUser::StopAssociation ()`

Stop the running association.

Examples:

`CStoreQtProgress.cxx`.

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

- `gdcmServiceClassUser.h`

25.228 gdcm::SHA1 Class Reference

Class for SHA1.

```
#include <gdcmSHA1.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])`

25.228.1 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

25.228.2 Constructor & Destructor Documentation

25.228.2.1 `gdcm::SHA1::SHA1 ()`

25.228.2.2 `gdcm::SHA1::~~SHA1 ()`

25.228.3 Member Function Documentation

25.228.3.1 `static bool gdcM::SHA1::Compute (const char * buffer, unsigned long buf_len, char digest_str[20*2+1])`
[static]

25.228.3.2 `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:

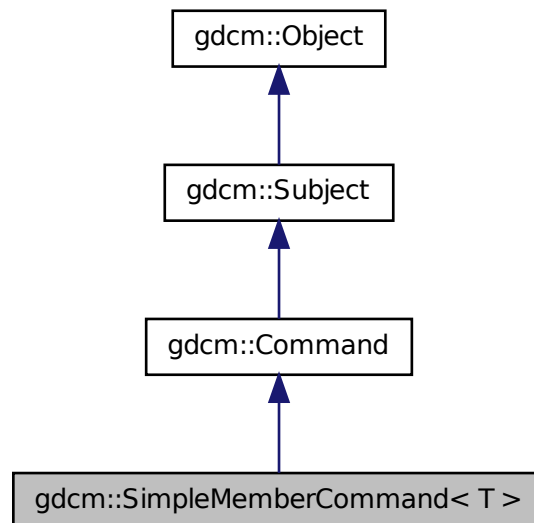
- gdcMSHA1.h

25.229 gdcM::SimpleMemberCommand< T > Class Template Reference

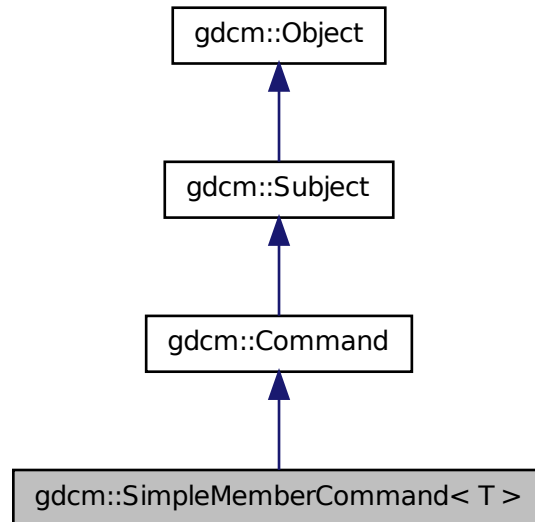
Command subclass that calls a pointer to a member function.

```
#include <gdcMCommand.h>
```

Inheritance diagram for gdcM::SimpleMemberCommand< T >:



Collaboration diagram for gdcmm::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

25.229.1 Detailed Description

`template<typename T>class gdcM::SimpleMemberCommand< T >`

Command subclass that calls a pointer to a member function.

SimpleMemberCommand calls a pointer to a member function with no arguments.

25.229.2 Member Typedef Documentation

25.229.2.1 `template<typename T > typedef SimpleMemberCommand gdcM::SimpleMemberCommand< T >::Self`

Standard class typedefs.

25.229.2.2 `template<typename T > typedef void(T::* gdcM::SimpleMemberCommand< T >::TMemberFunctionPointer)()`

A method callback.

25.229.3 Constructor & Destructor Documentation

25.229.3.1 `template<typename T > gdcM::SimpleMemberCommand< T >::SimpleMemberCommand ()`
`[inline], [protected]`

Referenced by `gdcM::SimpleMemberCommand< T >::New()`.

25.229.3.2 `template<typename T > virtual gdcM::SimpleMemberCommand< T >::~~SimpleMemberCommand ()`
`[inline], [protected], [virtual]`

25.229.4 Member Function Documentation

25.229.4.1 `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`.

25.229.4.2 `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.

25.229.4.3 `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().

25.229.4.4 `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.

25.229.5 Member Data Documentation

25.229.5.1 `template<typename T > TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction [protected]`

Referenced by gdcm::SimpleMemberCommand< T >::Execute(), and gdcm::SimpleMemberCommand< T >::SetCallbackFunction().

25.229.5.2 `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

25.230 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 ShowIteration ()
- virtual void ShowProgress (Subject *caller, const Event &evt)
- virtual void StartFilter ()
- void TestAbortOff ()
- void TestAbortOn ()

25.230.1 Detailed Description

SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events.

25.230.2 Constructor & Destructor Documentation

25.230.2.1 `gdcmm::SimpleSubjectWatcher::SimpleSubjectWatcher (Subject * s, const char * comment = " ")`

25.230.2.2 `virtual gdcmm::SimpleSubjectWatcher::~SimpleSubjectWatcher ()` [virtual]

25.230.3 Member Function Documentation

25.230.3.1 `virtual void gdcmm::SimpleSubjectWatcher::EndFilter ()` [protected],[virtual]

25.230.3.2 `virtual void gdcmm::SimpleSubjectWatcher::ShowAbort ()` [protected],[virtual]

25.230.3.3 `virtual void gdcmm::SimpleSubjectWatcher::ShowAnonymization (Subject * caller, const Event & evt)` [protected],[virtual]

25.230.3.4 `virtual void gdcmm::SimpleSubjectWatcher::ShowData (Subject * caller, const Event & evt)` [protected],[virtual]

25.230.3.5 `virtual void gdcmm::SimpleSubjectWatcher::ShowDataSet (Subject * caller, const Event & evt)` [protected],[virtual]

25.230.3.6 `virtual void gdcmm::SimpleSubjectWatcher::ShowIteration ()` [protected],[virtual]

25.230.3.7 `virtual void gdcmm::SimpleSubjectWatcher::ShowProgress (Subject * caller, const Event & evt)` [protected],[virtual]

25.230.3.8 `virtual void gdcmm::SimpleSubjectWatcher::StartFilter ()` [protected],[virtual]

25.230.3.9 `void gdcmm::SimpleSubjectWatcher::TestAbortOff ()` [protected]

25.230.3.10 `void gdcmm::SimpleSubjectWatcher::TestAbortOn ()` [protected]

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

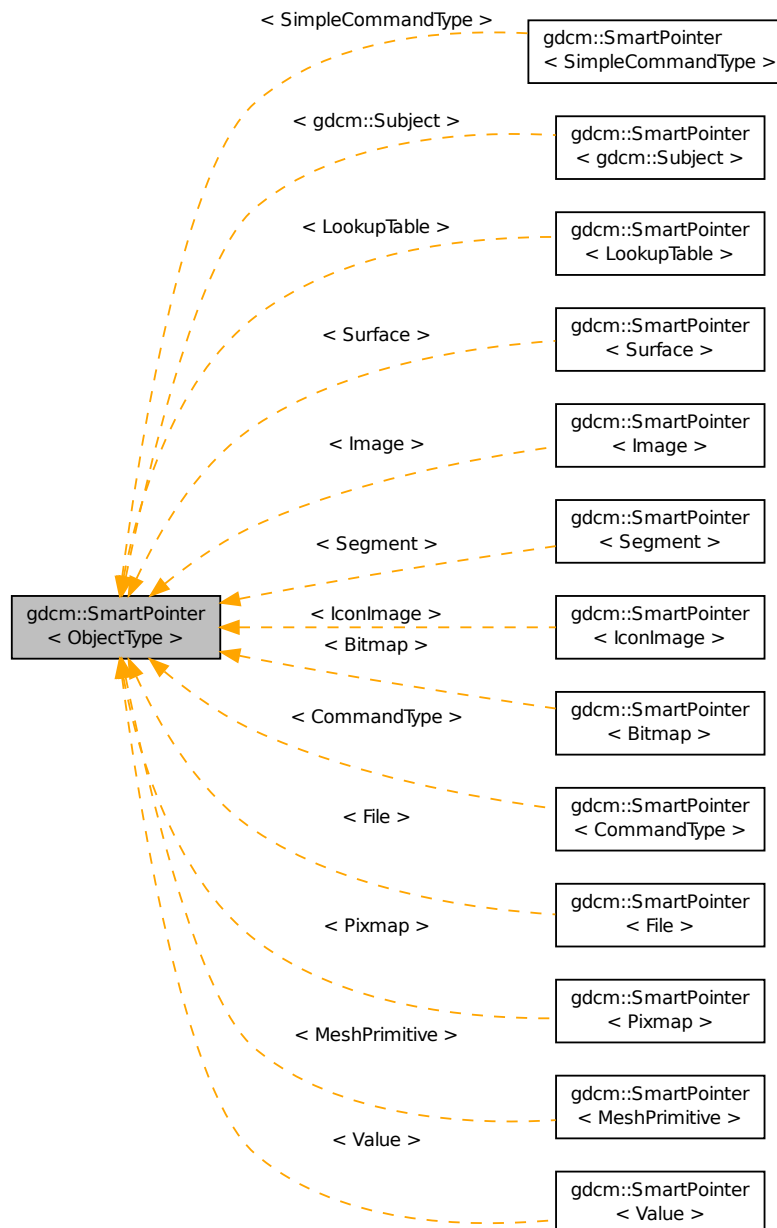
- `gdcmmSimpleSubjectWatcher.h`

25.231 gdcM::SmartPointer< ObjectType > Class Template Reference

Class for Smart Pointer.

```
#include <gdcMSmartPointer.h>
```

Inheritance diagram for gdcM::SmartPointer< ObjectType >:



Public Member Functions

- `SmartPointer ()`
- `SmartPointer (const SmartPointer< ObjectType > &p)`
- `SmartPointer (ObjectType *p)`
- `SmartPointer (ObjectType const &p)`
- `~SmartPointer ()`
- `ObjectType * GetPointer () const`

Explicit function to retrieve the pointer.

- `operator ObjectType * () const`

Return pointer to object.

- `ObjectType & operator* () const`
- `ObjectType * operator-> () const`

Overload operator ->

- `SmartPointer & operator= (SmartPointer const &r)`

Overload operator assignment.

- `SmartPointer & operator= (ObjectType *r)`

Overload operator assignment.

- `SmartPointer & operator= (ObjectType const &r)`

25.231.1 Detailed Description

```
template<class ObjectType>class gdcM::SmartPointer< ObjectType >
```

Class for Smart Pointer.

Will only work for subclass of `gdcM::Object` See `tr1/shared_ptr` for a more general approach (not invasive) `#include <tr1/memory> { shared_ptr<Bla> b(new Bla); }`

Note

Class partly based on post by Bill Hubauer: <http://groups.google.com/group/comp.lang.-c++/msg/173ddc38a827a930>

See also

<http://www.davethehat.com/articles/smarty.htm>

and `itk::SmartPointer`

Examples:

`ChangeSequenceUltrasound.cxx`, `CStoreQtProgress.cxx`, `DumpGEMSMovieGroup.cxx`, `Extracting_All_Resolution.cxx`, `Fake_Image_Using_Stream_Image_Writer.cxx`, `FixBrokenJ2K.cxx`, `gdcMrtionplan.cxx`, `gdcMrtplan.cxx`, `Gen-AllVR.cxx`, `GenFakelIdentifyFile.cxx`, `GenFakelImage.cxx`, `GenLongSeqs.cxx`, `GenSeqs.cxx`, `GetSubSequence-Data.cxx`, `LargeVRDSExplicit.cxx`, `ReadAndDumpDICOMDIR.cxx`, and `ReadExplicitLengthSQIVR.cxx`.

25.231.2 Constructor & Destructor Documentation

25.231.2.1 `template<class ObjectType> gdc::SmartPointer< ObjectType >::SmartPointer () [inline]`

25.231.2.2 `template<class ObjectType> gdc::SmartPointer< ObjectType >::SmartPointer (const SmartPointer< ObjectType > & p) [inline]`

25.231.2.3 `template<class ObjectType> gdc::SmartPointer< ObjectType >::SmartPointer (ObjectType * p) [inline]`

25.231.2.4 `template<class ObjectType> gdc::SmartPointer< ObjectType >::SmartPointer (ObjectType const & p) [inline]`

25.231.2.5 `template<class ObjectType> gdc::SmartPointer< ObjectType >::~~SmartPointer () [inline]`

25.231.3 Member Function Documentation

25.231.3.1 `template<class ObjectType> ObjectType* gdc::SmartPointer< ObjectType >::GetPointer () const [inline]`

Explicit function to retrieve the pointer.

25.231.3.2 `template<class ObjectType> gdc::SmartPointer< ObjectType >::operator ObjectType * () const [inline]`

Return pointer to object.

25.231.3.3 `template<class ObjectType> ObjectType& gdc::SmartPointer< ObjectType >::operator* () const [inline]`

25.231.3.4 `template<class ObjectType> ObjectType* gdc::SmartPointer< ObjectType >::operator-> () const [inline]`

Overload operator ->

25.231.3.5 `template<class ObjectType> SmartPointer& gdc::SmartPointer< ObjectType >::operator= (SmartPointer< ObjectType > const & r) [inline]`

Overload operator assignment.

Referenced by `gdc::SmartPointer< Value >::operator=()`.

25.231.3.6 `template<class ObjectType> SmartPointer& gdc::SmartPointer< ObjectType >::operator= (ObjectType * r) [inline]`

Overload operator assignment.

```
25.231.3.7 template<class ObjectType> SmartPointer& gdcM::SmartPointer< ObjectType >::operator= ( ObjectType const
& r ) [inline]
```

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

- gdcMSmartPointer.h

25.232 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 ()

25.232.1 Detailed Description

Class convert a class SOP Class UID into IOD.

Reference PS 3.4 Table B.5-1 STANDARD SOP CLASSES

25.232.2 Member Typedef Documentation

```
25.232.2.1 typedef const char* gdcM::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]
```

25.232.3 Member Function Documentation

```
25.232.3.1 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.
```

25.232.3.2 `static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (const char * sopclassuid)` [static]

25.232.3.3 `static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ()` [static]

Return the number of SOP Class UID listed internally.

25.232.3.4 `static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (const char * iod)` [static]

25.232.3.5 `static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (unsigned int i)` [static]

25.232.3.6 `static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ()` [static]

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

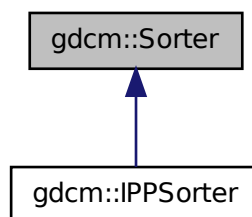
- `gdcmSOPClassUIDToIOD.h`

25.233 gdcm::Sorter Class Reference

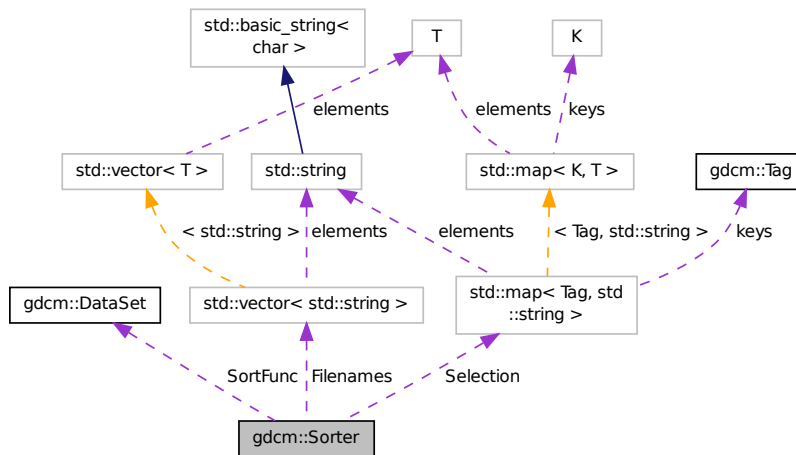
Sorter General class to do sorting using a custom function You simply need to provide a function of type: `Sorter::Sort-Function`.

```
#include <gdcmSorter.h>
```

Inheritance diagram for `gdcm::Sorter`:



Collaboration diagram for `gdcm::Sorter`:



Public Types

- typedef `bool(* SortFunction)(DataSet const &, DataSet const &)`

Set the sort function which compares one dataset to the other.

Public Member Functions

- `Sorter()`
- `virtual ~Sorter()`
- `bool AddSelect(Tag const &tag, const char *value)`
UNSUPPORTED FOR NOW.
- `const std::vector< std::string > & GetFilenames() const`
- `void Print(std::ostream &os) const`
Print.
- `void SetSortFunction(SortFunction f)`
- `virtual bool Sort(std::vector< std::string > const &filenames)`
Typically the output of `gdcm::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)`

25.233.1 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.

25.233.2 Member Typedef Documentation

25.233.2.1 `typedef std::map<Tag, std::string> gdcm::Sorter::SelectionMap` `[protected]`

25.233.2.2 `typedef bool(* gdcm::Sorter::SortFunction)(DataSet const &, DataSet const &)`

Set the sort function which compares one dataset to the other.

25.233.3 Constructor & Destructor Documentation

25.233.3.1 `gdcm::Sorter::Sorter ()`

25.233.3.2 `virtual gdcm::Sorter::~~Sorter ()` `[virtual]`

25.233.4 Member Function Documentation

25.233.4.1 `bool gdcm::Sorter::AddSelect (Tag const & tag, const char * value)`

UNSUPPORTED FOR NOW.

25.233.4.2 `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:

gdcmmorthoplanes.cxx, reslicesphere.cxx, SortImage.cxx, and VolumeSorter.cxx.

25.233.4.3 `void gdcmm::Sorter::Print (std::ostream & os) const`

Print.

Examples:

gdcmmorthoplanes.cxx, SortImage.cxx, and VolumeSorter.cxx.

Referenced by gdcmm::operator<<().

25.233.4.4 `void gdcmm::Sorter::SetSortFunction (SortFunction f)`

Examples:

SortImage.cxx, and VolumeSorter.cxx.

25.233.4.5 `virtual bool gdcmm::Sorter::Sort (std::vector< std::string > const & filenames)` `[virtual]`

Typically the output of gdcmm::Directory::GetFileNames()

Reimplemented in gdcmm::IPPSorter.

Examples:

SortImage.cxx.

25.233.4.6 `virtual bool gdcmm::Sorter::StableSort (std::vector< std::string > const & filenames)` `[virtual]`

Examples:

SortImage.cxx, and VolumeSorter.cxx.

25.233.5 Friends And Related Function Documentation

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

25.233.6 Member Data Documentation

25.233.6.1 `std::vector<std::string> gdcmm::Sorter::FileNames` `[protected]`

25.233.6.2 `std::map<Tag, std::string> gdcmm::Sorter::Selection` `[protected]`

25.233.6.3 SortFunction gdcm::Sorter::SortFunc [protected]

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

- gdcmSorter.h

25.234 gdcm::Spacing Class Reference

Class for Spacing.

```
#include <gdcmSpacing.h>
```

Public Types

- enum SpacingType {
DETECTOR = 0,
MAGNIFIED,
CALIBRATED,
UNKNOWN }

Public Member Functions

- Spacing ()
- ~Spacing ()

Static Public Member Functions

- static Attribute< 0x28, 0x34 > ComputePixelAspectRatioFromPixelSpacing (const Attribute< 0x28, 0x30 > &pixelspacing)

25.234.1 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://apps.sourceforge.net/mediawiki/gdcm/index.php?title=Imager_Pixel_Spacing

25.234.2 Member Enumeration Documentation

25.234.2.1 enum gdcm::Spacing::SpacingType

Enumerator:

DETECTOR

MAGNIFIED

CALIBRATED

UNKNOWN

25.234.3 Constructor & Destructor Documentation

25.234.3.1 gdcm::Spacing::Spacing ()

25.234.3.2 gdcm::Spacing::~~Spacing ()

25.234.4 Member Function Documentation

25.234.4.1 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

25.235 gdcm::Spectroscopy Class Reference

Spectroscopy class.

```
#include <gdcmSpectroscopy.h>
```

Public Member Functions

- Spectroscopy ()

25.235.1 Detailed Description

Spectroscopy class.

25.235.2 Constructor & Destructor Documentation

25.235.2.1 gdcm::Spectroscopy::Spectroscopy () [inline]

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

- gdcmSpectroscopy.h

25.236 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.

25.236.1 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.

25.236.2 Constructor & Destructor Documentation

25.236.2.1 `gdcm::SplitMosaicFilter::SplitMosaicFilter ()`

25.236.2.2 `gdcm::SplitMosaicFilter::~~SplitMosaicFilter ()`

25.236.3 Member Function Documentation

25.236.3.1 `bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (unsigned int dims[3])`

Compute the new dimensions according to private information stored in the MOSAIC header.

25.236.3.2 `File& gdcm::SplitMosaicFilter::GetFile ()` `[inline]`

25.236.3.3 `const File& gdcm::SplitMosaicFilter::GetFile () const` `[inline]`

25.236.3.4 `const Image& gdcm::SplitMosaicFilter::GetImage () const` `[inline]`

25.236.3.5 `Image& gdcm::SplitMosaicFilter::GetImage ()` `[inline]`

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

25.236.3.7 `void gdcm::SplitMosaicFilter::SetImage (const Image & image)`

25.236.3.8 `bool gdcm::SplitMosaicFilter::Split ()`

Split the SIEMENS MOSAIC image.

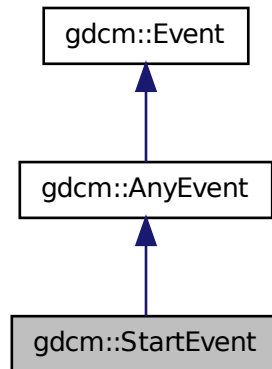
The documentation for this class was generated from the following file:

- `gdcmSplitMosaicFilter.h`

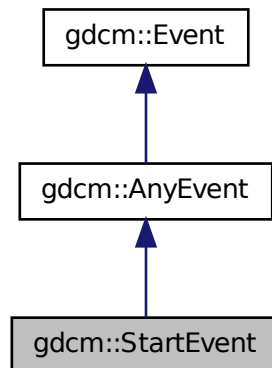
25.237 gdcm::StartEvent Class Reference

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

Inheritance diagram for `gdcm::StartEvent`:



Collaboration diagram for `gdcm::StartEvent`:



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

- `gdcmEvent.h`

25.238 `gdcm::static_assert_test< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- `gdcmStaticAssert.h`

25.239 `gdcm::STATIC_ASSERTION_FAILURE< true >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

Public Types

- `enum { value = 1 }`

25.239.1 Member Enumeration Documentation

25.239.1.1 anonymous enum

Enumerator:

value

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

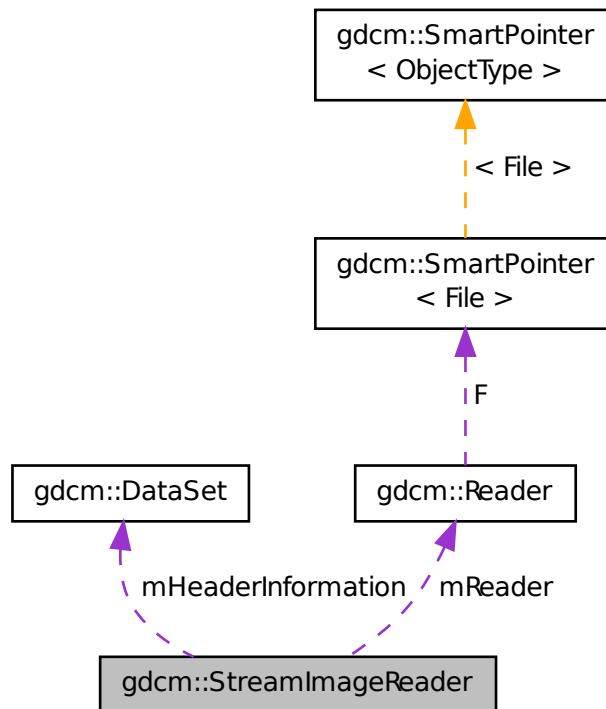
- `gdcmStaticAssert.h`

25.240 `gdcm::StreamImageReader` Class Reference

`StreamImageReader`.

```
#include <gdcmStreamImageReader.h>
```

Collaboration diagram for gdcm::StreamImageReader:



Public Member Functions

- `StreamImageReader ()`
- `~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 (void *inReadBuffer, const std::size_t &inBufferLength)`
- `virtual bool ReadImageInformation ()`
- `void SetFileName (const char *inFileName)`
- `void SetStream (std::istream &inStream)`

Protected Member Functions

- `bool ReadImageSubregionJpegLS (char *inReadBuffer, const std::size_t &inBufferLength)`
- `virtual bool ReadImageSubregionRAW (char *inReadBuffer, const std::size_t &inBufferLength)`

Protected Attributes

- `std::streamoff mFileOffset`
- `std::streamoff mFileOffset1`
- `DataSet mHeaderInformation`
- `Reader mReader`
- `uint16_t mXMax`
- `uint16_t mXMin`
- `uint16_t mYMax`
- `uint16_t mYMin`
- `uint16_t mZMax`
- `uint16_t mZMin`

25.240.1 Detailed Description

`StreamImageReader`.

Note

its role is to convert the DICOM `DataSet` into a `gdcm::Image` representation via an ITK streaming (ie, multithreaded) interface `Image` is different from `Pixmap` as 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`.

25.240.2 Constructor & Destructor Documentation

25.240.2.1 `gdcm::StreamImageReader::StreamImageReader ()`

25.240.2.2 `gdcm::StreamImageReader::~~StreamImageReader ()`

25.240.3 Member Function Documentation

25.240.3.1 `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`.

25.240.3.2 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. 15 nov 2010: added z dimension, defaults to being 1 plane large

Examples:

StreamImageReaderTest.cxx.

25.240.3.3 uint32_t gdcm::StreamImageReader::DefineProperBufferLength () const

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. Call this function to determine the size of the char* buffer that will need to be passed in to ReadImageSubregion(). If the return is 0, then that means that the pixel extent was not defined prior

Examples:

StreamImageReaderTest.cxx.

25.240.3.4 std::vector<unsigned int> gdcm::StreamImageReader::GetDimensionsValueForResolution (unsigned int)

25.240.3.5 File const& gdcm::StreamImageReader::GetFile () const

Returns the dataset read by ReadImageInformation Couple this with the ImageHelper to get statistics about the image, like pixel extent, to be able to initialize buffers for reading

Examples:

StreamImageReaderTest.cxx.

25.240.3.6 bool gdcm::StreamImageReader::Read (void * *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 void* to std::ostream (internally) fails
3. the given buffer isn't large enough to accomodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use ImageReader instead.

Examples:

StreamImageReaderTest.cxx.

25.240.3.7 `virtual bool gdcm::StreamImageReader::ReadImageInformation () [virtual]`

Set the spacing and dimension information for the set filename. returns false if the file is not initialized or not an image, with the pixel 0x7fe0, 0x0010 tag.

Examples:

StreamImageReaderTest.cxx.

25.240.3.8 `bool gdcm::StreamImageReader::ReadImageSubregionJpegLS (char * inReadBuffer, const std::size_t & inBufferLength) [protected]`

Reads the file via JpegLS. The JpegLS codec, as of this writing, requires that the entire file be read in in order to decode a subregion, so that's what's done here.

25.240.3.9 `virtual bool gdcm::StreamImageReader::ReadImageSubregionRAW (char * inReadBuffer, 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

25.240.3.10 `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:

StreamImageReaderTest.cxx.

25.240.3.11 `void gdcm::StreamImageReader::SetStream (std::istream & inStream)`

25.240.4 Member Data Documentation

25.240.4.1 `std::streamoff gdcm::StreamImageReader::mFileOffset [protected]`

25.240.4.2 `std::streamoff gdcm::StreamImageReader::mFileOffset1 [protected]`

25.240.4.3 `DataSet gdcm::StreamImageReader::mHeaderInformation [protected]`

25.240.4.4 `Reader gdcm::StreamImageReader::mReader [protected]`

25.240.4.5 `uint16_t gdcm::StreamImageReader::mXMax [protected]`

25.240.4.6 `uint16_t gdcm::StreamImageReader::mXMin [protected]`

25.240.4.7 `uint16_t gdcm::StreamImageReader::mYMax [protected]`

25.240.4.8 `uint16_t gdcm::StreamImageReader::mYMin [protected]`

25.240.4.9 `uint16_t gdcm::StreamImageReader::mZMax` [protected]

25.240.4.10 `uint16_t gdcm::StreamImageReader::mZMin` [protected]

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

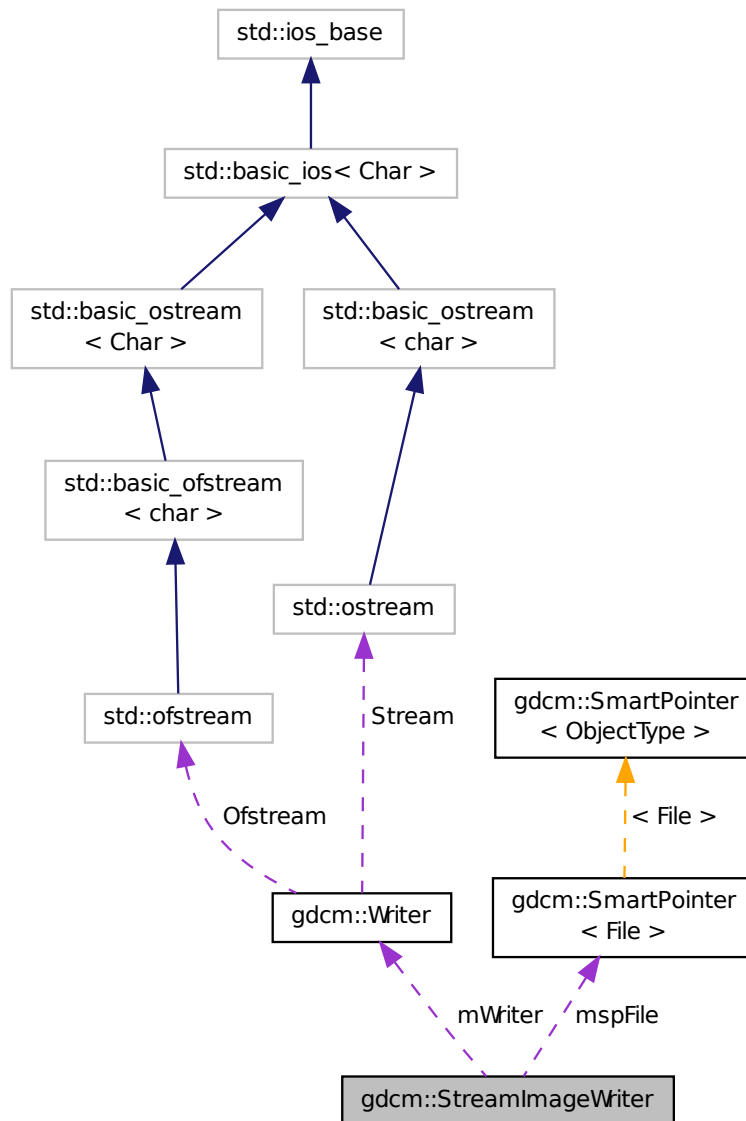
- `gdcmStreamImageReader.h`

25.241 gdcm::StreamImageWriter Class Reference

StreamImageReader.

```
#include <gdcmStreamImageWriter.h>
```

Collaboration diagram for `gdcm::StreamImageWriter`:



Public Member Functions

- `StreamImageWriter ()`
- `~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

25.241.1 Detailed Description

StreamImageReader.

Note

its role is to convert the DICOM DataSet into a gdcm::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:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

25.241.2 Constructor & Destructor Documentation

25.241.2.1 gdcm::StreamImageWriter::StreamImageWriter ()

25.241.2.2 gdcm::StreamImageWriter::~~StreamImageWriter ()

25.241.3 Member Function Documentation

25.241.3.1 `bool gdcmm::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`.

25.241.3.2 `void gdcmm::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`.

25.241.3.3 `uint32_t gdcmm::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`.

25.241.3.4 `void gdcmm::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`.

25.241.3.5 `void gdcmm::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.

25.241.3.6 `void gdcmm::StreamImageWriter::SetStream (std::ostream & inStream)`

Examples:

`Extracting_All_Resolution.cxx`, `Fake_Image_Using_Stream_Image_Writer.cxx`, and `StreamImageReaderTest.cxx`.

25.241.3.7 `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.

25.241.3.8 `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.

25.241.3.9 `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

25.241.3.10 `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.

25.241.4 Member Data Documentation

25.241.4.1 `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.

- 25.241.4.2 `int gdcM::StreamImageWriter::mElementOffsets1` [protected]
- 25.241.4.3 `SmartPointer<File> gdcM::StreamImageWriter::mspFile` [protected]
- 25.241.4.4 `Writer gdcM::StreamImageWriter::mWriter` [protected]
- 25.241.4.5 `uint16_t gdcM::StreamImageWriter::mXMax` [protected]
- 25.241.4.6 `uint16_t gdcM::StreamImageWriter::mXMin` [protected]
- 25.241.4.7 `uint16_t gdcM::StreamImageWriter::mYMax` [protected]
- 25.241.4.8 `uint16_t gdcM::StreamImageWriter::mYMin` [protected]
- 25.241.4.9 `uint16_t gdcM::StreamImageWriter::mZMax` [protected]
- 25.241.4.10 `uint16_t gdcM::StreamImageWriter::mZMin` [protected]

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

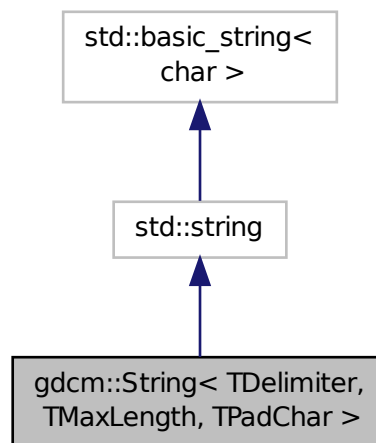
- `gdcMStreamImageWriter.h`

25.242 `gdcM::String< TDelimiter, TMaxLength, TPadChar >` Class Template Reference

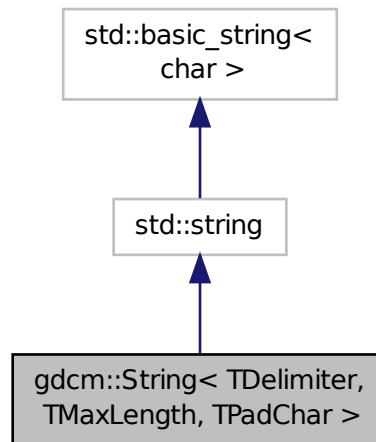
`String`.

```
#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=npow)`
- `bool IsValid () const`
return if string is valid

- operator const char * () const
WARNING: Trailing \0 might be lost in this operation:
- std::string Trim () const
- gdcmm::String< TDelimiter,
TMaxLength, TPadChar > Truncate () const

25.242.1 Detailed Description

template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>class gdcmm::String< TDelimiter, TMaxLength, TPadChar >

String.

Note

TDelimiter template parameter is used to separate multiple String (VM1 >) TMaxLength is only a hint. Noone actually respect the max length TPadChar is the string padding (0 or space)

25.242.2 Member Typedef Documentation

- 25.242.2.1 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_iterator
- 25.242.2.2 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reference gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_reference
- 25.242.2.3 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reverse_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_reverse_iterator
- 25.242.2.4 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::difference_type gdcmm::String< TDelimiter, TMaxLength, TPadChar >::difference_type
- 25.242.2.5 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::iterator
- 25.242.2.6 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::pointer gdcmm::String< TDelimiter, TMaxLength, TPadChar >::pointer
- 25.242.2.7 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reference gdcmm::String< TDelimiter, TMaxLength, TPadChar >::reference
- 25.242.2.8 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reverse_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::reverse_iterator
- 25.242.2.9 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::size_type gdcmm::String< TDelimiter, TMaxLength, TPadChar >::size_type
- 25.242.2.10 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::value_type gdcmm::String< TDelimiter, TMaxLength, TPadChar >::value_type

25.242.3 Constructor & Destructor Documentation

25.242.3.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '>'> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String () [inline]`

String constructors.

25.242.3.2 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '>'> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * s) [inline]`

25.242.3.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '>'> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * s, size_type n) [inline]`

25.242.3.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '>'> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const std::string & s, size_type pos = 0, size_type n = npos) [inline]`

25.242.4 Member Function Documentation

25.242.4.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '>'> bool gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid () const [inline]`

return if string is valid

Referenced by `gdcm::LO::IsValid()`, and `gdcm::String< TDelimiter, TMaxLength, TPadChar >::Truncate()`.

25.242.4.2 `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:

25.242.4.3 `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...

25.242.4.4 `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`

25.243 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)
DEPRECATED: NEVER USE IT.
- std::string FromString (const Tag &t, const char *value, size_t 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 Tag &t) const
Convert to string the ByteValue contained in a DataElement.
- std::pair< std::string,
std::string > ToStringPair (const Tag &t) const
- 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

25.243.1 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.

25.243.2 Constructor & Destructor Documentation

25.243.2.1 gdc2::StringFilter::StringFilter ()

25.243.2.2 gdc2::StringFilter::~~StringFilter ()

25.243.3 Member Function Documentation

25.243.3.1 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

25.243.3.2 `bool gdcm::StringFilter::ExecuteQuery (std::string const & query, DataSet const & ds, std::string & value) const`
`[protected]`

25.243.3.3 `std::string gdcm::StringFilter::FromString (const Tag & t, const char * value, VL const & vl)`

DEPRECATED: NEVER USE IT.

25.243.3.4 `std::string gdcm::StringFilter::FromString (const Tag & t, const char * value, size_t len)`

25.243.3.5 `File& gdcm::StringFilter::GetFile () [inline]`

25.243.3.6 `const File& gdcm::StringFilter::GetFile () const [inline]`

25.243.3.7 `void gdcm::StringFilter::SetDicts (const Dicts & dicts)`

Allow user to pass in there own dicts.

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

Set/Get File.

Examples:

`ReadAndPrintAttributes.cxx.`

25.243.3.9 `std::string gdcm::StringFilter::ToString (const Tag & t) const`

Convert to string the ByteValue contained in a DataElement.

Examples:

`ReadAndPrintAttributes.cxx.`

25.243.3.10 `std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & t) 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.`

25.243.3.11 `std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & t, DataSet const & ds) const`
`[protected]`

25.243.3.12 `void gdcm::StringFilter::UseDictAlways (bool) [inline]`

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

- `gdcmStringFilter.h`

25.244 gdcm::Study Class Reference

Study.

```
#include <gdcmStudy.h>
```

Public Member Functions

- Study ()

25.244.1 Detailed Description

Study.

25.244.2 Constructor & Destructor Documentation

25.244.2.1 gdcm::Study::Study () [inline]

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

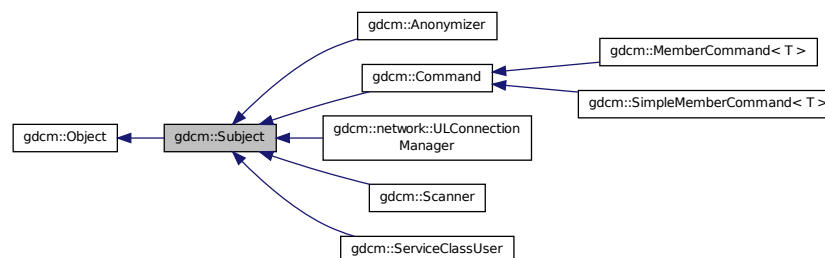
- gdcmStudy.h

25.245 gdcm::Subject Class Reference

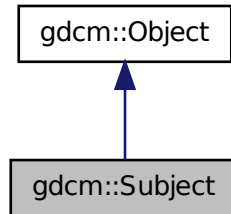
Subject.

```
#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

25.245.1 Detailed Description

Subject.

See also

Command Event

25.245.2 Constructor & Destructor Documentation

25.245.2.1 `gdcm::Subject::Subject ()`

25.245.2.2 `gdcm::Subject::~~Subject ()`

25.245.3 Member Function Documentation

25.245.3.1 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

25.245.3.2 unsigned long gdcM::Subject::AddObserver (const Event & event, Command *) const

25.245.3.3 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.

25.245.3.4 bool gdcM::Subject::HasObserver (const Event & event) const

Return true if an observer is registered for this event.

25.245.3.5 void gdcM::Subject::InvokeEvent (const Event &)

Call Execute on all the Commands observing this event id.

25.245.3.6 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.

25.245.3.7 void gdcM::Subject::RemoveAllObservers ()

Remove all observers .

25.245.3.8 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:

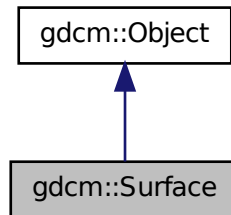
- gdcMSubject.h

25.246 gdcM::Surface Class Reference

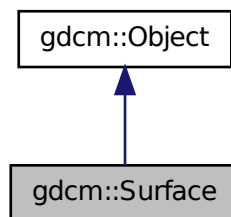
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

```
#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

25.246.1 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

25.246.2 Member Enumeration Documentation

25.246.2.1 enum gdcm::Surface::STATES

Enumerator:

NO

YES

UNKNOWN

STATES_END

25.246.2.2 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

25.246.3 Constructor & Destructor Documentation

25.246.3.1 gdcm::Surface::Surface ()

25.246.3.2 virtual gdcm::Surface::~~Surface () [virtual]

25.246.4 Member Function Documentation

25.246.4.1 SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetAlgorithmFamily () const

25.246.4.2 SegmentHelper::BasicCodedEntry& gdcm::Surface::GetAlgorithmFamily ()

25.246.4.3 const char* gdcm::Surface::GetAlgorithmName () const

25.246.4.4 const char* gdcm::Surface::GetAlgorithmVersion () const

25.246.4.5 const float* gdcm::Surface::GetAxisOfRotation () const

Note

Pointer is null if undefined

25.246.4.6 const float* gdcm::Surface::GetCenterOfRotation () const

Note

Pointer is null if undefined

25.246.4.7 STATES gdcm::Surface::GetFiniteVolume () const

25.246.4.8 STATES gdcm::Surface::GetManifold () const

25.246.4.9 float gdcm::Surface::GetMaximumPointDistance () const

25.246.4.10 float gdcm::Surface::GetMeanPointDistance () const

25.246.4.11 **MeshPrimitive** const& gdcm::Surface::GetMeshPrimitive () const

25.246.4.12 **MeshPrimitive&** gdcm::Surface::GetMeshPrimitive ()

25.246.4.13 unsigned long gdcm::Surface::GetNumberOfSurfacePoints () const

25.246.4.14 unsigned long gdcm::Surface::GetNumberOfVectors () const

25.246.4.15 const **DataElement&** gdcm::Surface::GetPointCoordinatesData () const

25.246.4.16 **DataElement&** gdcm::Surface::GetPointCoordinatesData ()

25.246.4.17 const float* gdcm::Surface::GetPointPositionAccuracy () const

Note

Pointer is null if undefined

25.246.4.18 const float* gdcm::Surface::GetPointsBoundingBoxCoordinates () const

Note

Pointer is null if undefined

25.246.4.19 **SegmentHelper::BasicCodedEntry** const& gdcm::Surface::GetProcessingAlgorithm () const

25.246.4.20 **SegmentHelper::BasicCodedEntry&** gdcm::Surface::GetProcessingAlgorithm ()

25.246.4.21 const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue () const

25.246.4.22 unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (const unsigned int *idx*) const

25.246.4.23 unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue () const

25.246.4.24 float gdcm::Surface::GetRecommendedPresentationOpacity () const

25.246.4.25 **VIEWType** gdcm::Surface::GetRecommendedPresentationType () const

25.246.4.26 static **STATES** gdcm::Surface::GetSTATES (const char * *state*) [static]

25.246.4.27 static const char* gdcm::Surface::GetSTATESString (**STATES** *state*) [static]

25.246.4.28 const char* gdcm::Surface::GetSurfaceComments () const

25.246.4.29 unsigned long gdcm::Surface::GetSurfaceNumber () const

25.246.4.30 bool gdcm::Surface::GetSurfaceProcessing () const

25.246.4.31 const char* gdcm::Surface::GetSurfaceProcessingDescription () const

25.246.4.32 float gdcm::Surface::GetSurfaceProcessingRatio () const

- 25.246.4.33 `const float* gdcmm::Surface::GetVectorAccuracy () const`
- 25.246.4.34 `const DataElement& gdcmm::Surface::GetVectorCoordinateData () const`
- 25.246.4.35 `DataElement& gdcmm::Surface::GetVectorCoordinateData ()`
- 25.246.4.36 `unsigned short gdcmm::Surface::GetVectorDimensionality () const`
- 25.246.4.37 `static VIEWType gdcmm::Surface::GetVIEWType (const char * type) [static]`
- 25.246.4.38 `static const char* gdcmm::Surface::GetVIEWTypeString (VIEWType type) [static]`
- 25.246.4.39 `void gdcmm::Surface::SetAlgorithmFamily (SegmentHelper::BasicCodedEntry const & BSE)`
- 25.246.4.40 `void gdcmm::Surface::SetAlgorithmName (const char * str)`
- 25.246.4.41 `void gdcmm::Surface::SetAlgorithmVersion (const char * str)`
- 25.246.4.42 `void gdcmm::Surface::SetAxisOfRotation (const float * axis)`
- 25.246.4.43 `void gdcmm::Surface::SetCenterOfRotation (const float * center)`
- 25.246.4.44 `void gdcmm::Surface::SetFiniteVolume (STATES state)`
- 25.246.4.45 `void gdcmm::Surface::SetManifold (STATES state)`
- 25.246.4.46 `void gdcmm::Surface::SetMaximumPointDistance (float maximum)`
- 25.246.4.47 `void gdcmm::Surface::SetMeanPointDistance (float average)`
- 25.246.4.48 `void gdcmm::Surface::SetMeshPrimitive (MeshPrimitive & mp)`
- 25.246.4.49 `void gdcmm::Surface::SetNumberOfSurfacePoints (const unsigned long nb)`
- 25.246.4.50 `void gdcmm::Surface::SetNumberOfVectors (const unsigned long nb)`
- 25.246.4.51 `void gdcmm::Surface::SetPointCoordinatesData (DataElement const & de)`
- 25.246.4.52 `void gdcmm::Surface::SetPointPositionAccuracy (const float * accuracies)`
- 25.246.4.53 `void gdcmm::Surface::SetPointsBoundingBoxCoordinates (const float * coordinates)`
- 25.246.4.54 `void gdcmm::Surface::SetProcessingAlgorithm (SegmentHelper::BasicCodedEntry const & BSE)`
- 25.246.4.55 `void gdcmm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short vl[3])`
- 25.246.4.56 `void gdcmm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short vl, const unsigned int idx = 0)`
- 25.246.4.57 `void gdcmm::Surface::SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > & vl)`
- 25.246.4.58 `void gdcmm::Surface::SetRecommendedDisplayGrayscaleValue (const unsigned short vl)`

25.246.4.59 void gdcm::Surface::SetRecommendedPresentationOpacity (const float *opacity*)

25.246.4.60 void gdcm::Surface::SetRecommendedPresentationType (VIEWType *type*)

25.246.4.61 void gdcm::Surface::SetSurfaceComments (const char * *comment*)

25.246.4.62 void gdcm::Surface::SetSurfaceNumber (const unsigned long *nb*)

25.246.4.63 void gdcm::Surface::SetSurfaceProcessing (bool *b*)

25.246.4.64 void gdcm::Surface::SetSurfaceProcessingDescription (const char * *description*)

25.246.4.65 void gdcm::Surface::SetSurfaceProcessingRatio (const float *ratio*)

25.246.4.66 void gdcm::Surface::SetVectorAccuracy (const float * *accuracy*)

25.246.4.67 void gdcm::Surface::SetVectorCoordinateData (DataElement const & *de*)

25.246.4.68 void gdcm::Surface::SetVectorDimensionality (const unsigned short *dim*)

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

- gdcmSurface.h

25.247 gdcm::SurfaceHelper Class Reference

```
#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 rangeMax=255)
Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename U >
static std::vector< float > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U rangeMax=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 rangeMax=255)
Convert a RGB color into DICOM grayscale (ready to write).

25.247.1 Member Typedef Documentation

25.247.1.1 `typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray`

25.247.2 Member Function Documentation

25.247.2.1 `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>T</i>	Type of CIELab components.
<i>U</i>	Type of rangeMax value.

25.247.2.2 `template<typename U > static 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.
----------	-------------------------

25.247.2.3 `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.

25.247.2.4 `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:

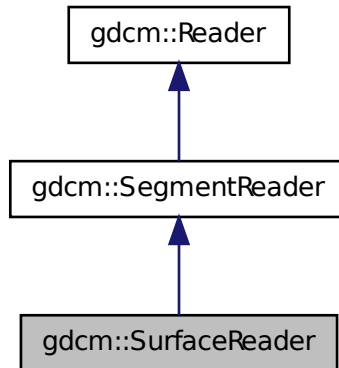
- gdcmSurfaceHelper.h

25.248 gdcm::SurfaceReader Class Reference

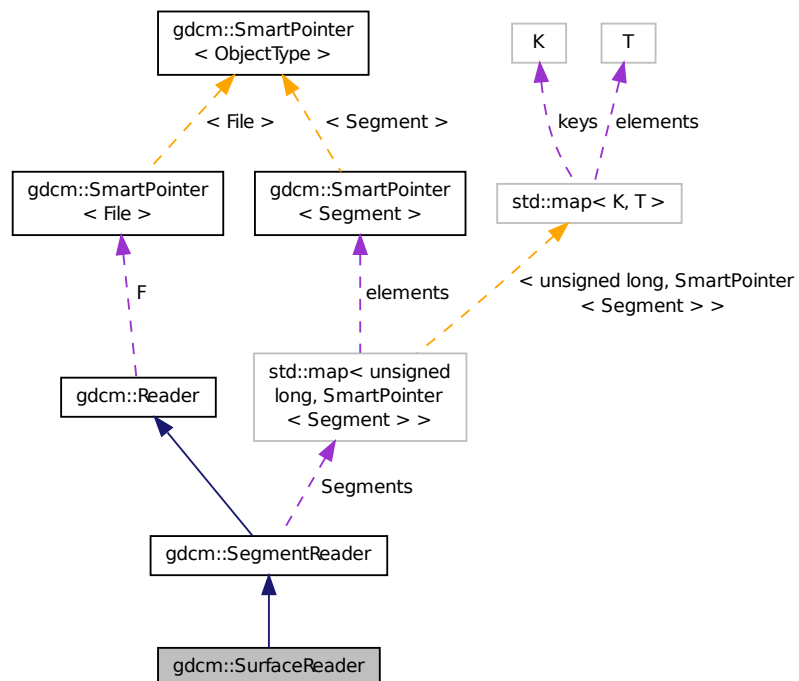
This class defines a SURFACE IE reader. It reads surface mesh module attributes.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for `gdc::SurfaceReader`:



Collaboration diagram for `gdc::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 &surfaceItem, const unsigned long idx)
- bool ReadSurfaces ()

Additional Inherited Members

25.248.1 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

25.248.2 Constructor & Destructor Documentation

25.248.2.1 gdcm::SurfaceReader::SurfaceReader ()

25.248.2.2 virtual gdcm::SurfaceReader::~~SurfaceReader () [virtual]

25.248.3 Member Function Documentation

25.248.3.1 unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces () const

25.248.3.2 virtual bool gdcm::SurfaceReader::Read () [virtual]

Read.

Reimplemented from gdcm::SegmentReader.

25.248.3.3 bool gdcm::SurfaceReader::ReadPointMacro (SmartPointer< Surface > surface, const DataSet & surfaceDS)
[protected]

25.248.3.4 bool gdcm::SurfaceReader::ReadSurface (const Item & surfaceItem, const unsigned long idx) [protected]

25.248.3.5 bool gdcm::SurfaceReader::ReadSurfaces () [protected]

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

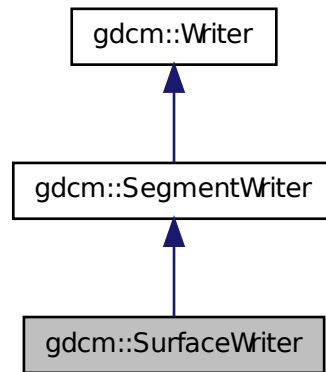
- gdcmSurfaceReader.h

25.249 gdcm::SurfaceWriter Class Reference

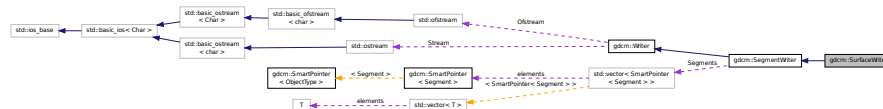
This class defines a SURFACE IE writer. It writes surface mesh module attributes.

```
#include <gdcmSurfaceWriter.h>
```

Inheritance diagram for gdcm::SurfaceWriter:



Collaboration diagram for gdcm::SurfaceWriter:



Public Member Functions

- `SurfaceWriter ()`
- `virtual ~SurfaceWriter ()`
- `unsigned long GetNumberOfSurfaces ()`
- `void SetNumberOfSurfaces (const unsigned long nb)`
- `bool Write ()`

Write.

Protected Member Functions

- `void ComputeNumberOfSurfaces ()`
- `bool PrepareWrite ()`
- `bool PrepareWritePointMacro (SmartPointer< Surface > surface, DataSet &surfaceDS, const TransferSyntax &ts)`

Protected Attributes

- unsigned long NumberOfSurfaces

Additional Inherited Members

25.249.1 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

25.249.2 Constructor & Destructor Documentation

25.249.2.1 `gdcm::SurfaceWriter::SurfaceWriter ()`

25.249.2.2 `virtual gdcm::SurfaceWriter::~~SurfaceWriter () [virtual]`

25.249.3 Member Function Documentation

25.249.3.1 `void gdcm::SurfaceWriter::ComputeNumberOfSurfaces () [protected]`

25.249.3.2 `unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ()`

25.249.3.3 `bool gdcm::SurfaceWriter::PrepareWrite () [protected]`

Reimplemented from `gdcm::SegmentWriter`.

25.249.3.4 `bool gdcm::SurfaceWriter::PrepareWritePointMacro (SmartPointer< Surface > surface, DataSet & surfaceDS, const TransferSyntax & ts) [protected]`

25.249.3.5 `void gdcm::SurfaceWriter::SetNumberOfSurfaces (const unsigned long nb)`

25.249.3.6 `bool gdcm::SurfaceWriter::Write () [virtual]`

Write.

Reimplemented from `gdcm::SegmentWriter`.

25.249.4 Member Data Documentation

25.249.4.1 `unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]`

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

- `gdcmSurfaceWriter.h`

25.250 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)

25.250.1 Detailed Description

SwapCode representation.

Examples:

TestByteSwap.cxx.

25.250.2 Member Enumeration Documentation

25.250.2.1 enum gdcm::SwapCode::SwapCodeType

Enumerator:

Unknown

LittleEndian

BigEndian

*BadLittleEndian**BadBigEndian*

25.250.3 Constructor & Destructor Documentation

25.250.3.1 `gdcm::SwapCode::SwapCode (SwapCodeType sc = Unknown)` `[inline]`

25.250.4 Member Function Documentation

25.250.4.1 `static int gdcm::SwapCode::GetIndex (SwapCode const & sc)` `[static]`, `[protected]`

25.250.4.2 `static const char* gdcm::SwapCode::GetSwapCodeString (SwapCode const & sc)` `[static]`

Referenced by `gdcm::operator<<()`.

25.250.4.3 `gdcm::SwapCode::operator SwapCode::SwapCodeType () const` `[inline]`

25.250.5 Friends And Related Function Documentation

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

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

- `gdcmSwapCode.h`

25.251 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)`

25.251.1 Member Function Documentation

25.251.1.1 `template<typename T > static T gdcm::SwapperDoOp::Swap (T val)` `[static]`

Referenced by `gdcm::Item::Read()`.

25.251.1.2 `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`

25.252 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)`

25.252.1 Detailed Description

Examples:

`ReadExplicitLengthSQIVR.cxx.`

25.252.2 Member Function Documentation

25.252.2.1 `template<typename T > static T gdcm::SwapperNoOp::Swap (T val)` `[inline], [static]`

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

25.252.2.2 `template<typename T > static void gdcm::SwapperNoOp::SwapArray (T *, size_t)` `[inline], [static]`

Referenced by `gdcm::EncodingImplementation< VR::VRBINARY >::Read()`.

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

- `gdcmSwapper.h`

25.253 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 * GetLastSystemError ()
 - 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 * 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)

25.253.1 Detailed Description

Class to do system operation.

OS independent functionalities

25.253.2 Member Function Documentation

25.253.2.1 static bool gdcm::System::DeleteDirectory (const char * *source*) [static]

remove a directory named source

25.253.2.2 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

25.253.2.3 `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.

25.253.2.4 `static bool gdcm::System::FilesDirectory (const char * name) [static]`

Check whether the file specified is a directory:

Examples:

gdcmorthoplanes.cxx, and threadgdcm.cxx.

25.253.2.5 `static bool gdcm::System::FilesSymlink (const char * name) [static]`

Check whether name is a symlink.

25.253.2.6 `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, and EncapsulateFileInRawData.cxx.

25.253.2.7 `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

25.253.2.8 `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]

25.253.2.9 `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

25.253.2.10 `static const char* gdcmm::System::GetCurrentModuleFileName () [static]`

Return the directory the current module is located: NOT THREAD SAFE

25.253.2.11 `static const char* gdcmm::System::GetCurrentProcessFileName () [static]`

Return the directory the current process (executable) is located: NOT THREAD SAFE

25.253.2.12 `static const char* gdcmm::System::GetCurrentResourcesDirectory () [static]`

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

25.253.2.13 `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

25.253.2.14 `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

25.253.2.15 `static const char* gdcmm::System::GetLastError () [static]`

Return the last error.

25.253.2.16 `static const char* gdcmm::System::GetLocaleCharSet () [static]`

return locale charmap

25.253.2.17 `static bool gdcmm::System::GetPermissions (const char * file, unsigned short & mode) [static],
[protected]`

NOT THREAD SAFE.

25.253.2.18 `static const char* gdcmm::System::GetTimezoneOffsetFromUTC () [static]`

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

25.253.2.19 `static bool gdcmm::System::MakeDirectory (const char * path) [static]`

Create a directory name path.

25.253.2.20 `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)

25.253.2.21 `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

25.253.2.22 `static bool gdcmm::System::RemoveFile (const char * source) [static]`

remove a file named source

25.253.2.23 `static bool gdcmm::System::SetPermissions (const char * file, unsigned short mode) [static],
[protected]`

25.253.2.24 `static int gdcmm::System::StrCaseCmp (const char * s1, const char * s2) [static]`

consistent func for C99 spec of strcasecmp/strncasecmp

Referenced by gdcmm::PrivateTag::operator<().

25.253.2.25 `static int gdcmm::System::StrNCaseCmp (const char * s1, const char * s2, size_t n) [static]`

Precondition

`n != 0`

25.253.2.26 `static char* gdcmm::System::StrTokR (char * ptr, const char * sep, char ** end) [static]`

strtok_r

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

- gdcmmSystem.h

25.254 gdcmm::Table Class Reference

Table.

```
#include <gdcmmTable.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)

25.254.1 Detailed Description

Table.

25.254.2 Member Typedef Documentation

25.254.2.1 `typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry`

25.254.3 Constructor & Destructor Documentation

25.254.3.1 `gdcm::Table::Table ()` `[inline]`

25.254.3.2 `gdcm::Table::~~Table ()` `[inline]`

25.254.4 Member Function Documentation

25.254.4.1 `const TableEntry& gdcm::Table::GetTableEntry (const Tag & tag) const` `[inline]`

25.254.4.2 `void gdcm::Table::InsertEntry (Tag const & tag, TableEntry const & te)` `[inline]`

25.254.5 Friends And Related Function Documentation

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

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

- gdcmTable.h

25.255 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 ()

25.255.1 Detailed Description

TableEntry.

25.255.2 Constructor & Destructor Documentation

25.255.2.1 `gdcm::TableEntry::TableEntry (const char * attribute = 0, Type const & type = Type (), const char * des = 0)`
`[inline]`

25.255.2.2 `gdcm::TableEntry::~~TableEntry ()` `[inline]`

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

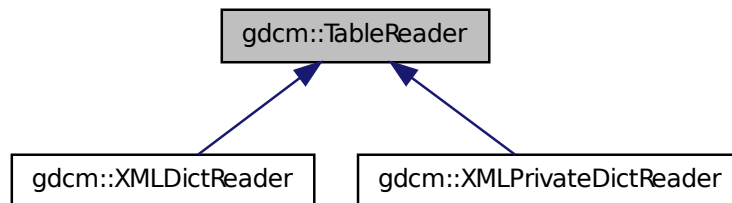
- `gdcmTableEntry.h`

25.256 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)

25.256.1 Detailed Description

Class for representing a TableReader.

Note

This class is an empty shell meant to be derived

25.256.2 Constructor & Destructor Documentation

25.256.2.1 gdcmm::TableReader::TableReader (Defs & defs) [inline]

25.256.2.2 virtual gdcmm::TableReader::~~TableReader () [inline],[virtual]

25.256.3 Member Function Documentation

25.256.3.1 virtual void gdcmm::TableReader::CharacterDataHandler (const char * data, int length) [virtual]

Reimplemented in gdcmm::XMLDictReader, and gdcmm::XMLPrivateDictReader.

25.256.3.2 virtual void gdcmm::TableReader::EndElement (const char * name) [virtual]

Reimplemented in gdcmm::XMLDictReader, and gdcmm::XMLPrivateDictReader.

25.256.3.3 const Defs& gdcmm::TableReader::GetDefs () const [inline]

25.256.3.4 const char* gdcmm::TableReader::GetFilename () [inline]

25.256.3.5 void gdcmm::TableReader::HandleIOD (const char ** atts)

25.256.3.6 void gdcmm::TableReader::HandleIODEntry (const char ** atts)

25.256.3.7 void gdcmm::TableReader::HandleMacro (const char ** atts)

25.256.3.8 void gdcmm::TableReader::HandleMacroEntry (const char ** atts)

25.256.3.9 void gdcmm::TableReader::HandleMacroEntryDescription (const char ** atts)

25.256.3.10 void gdcmm::TableReader::HandleModule (const char ** atts)

25.256.3.11 void gdcmm::TableReader::HandleModuleEntry (const char ** atts)

25.256.3.12 void gdcM::TableReader::HandleModuleEntryDescription (const char ** *atts*)

25.256.3.13 void gdcM::TableReader::HandleModuleInclude (const char ** *atts*)

25.256.3.14 int gdcM::TableReader::Read ()

25.256.3.15 void gdcM::TableReader::SetFilename (const char * *filename*) [inline]

25.256.3.16 virtual void gdcM::TableReader::StartElement (const char * *name*, const char ** *atts*) [virtual]

Reimplemented in gdcM::XMLDictReader, and gdcM::XMLPrivateDictReader.

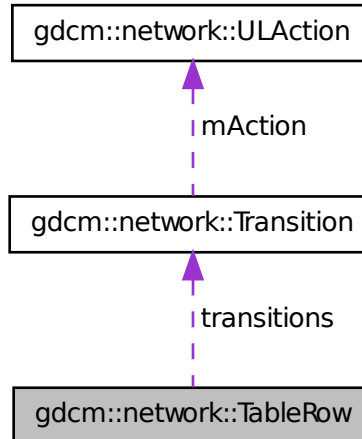
The documentation for this class was generated from the following file:

- gdcMTableReader.h

25.257 gdcM::network::TableRow Class Reference

```
#include <gdcMULTransitionTable.h>
```

Collaboration diagram for gdcM::network::TableRow:



Public Attributes

- Transition transitions [cMaxStateID]

25.257.1 Member Data Documentation

25.257.1.1 Transition gdcm::network::TableRow::transitions[cMaxStateID]

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

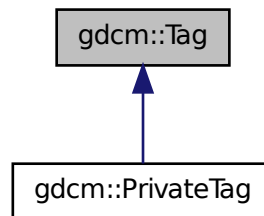
- gdcmULTransitionTable.h

25.258 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 PrintAsPipeSeparatedString () const`
 - `template<typename TSwap >`
`std::istream & Read (std::istream &is)`
Read a tag from binary representation.
 - `bool ReadFromCommaSeparatedString (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)`

25.258.1 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, CreateJPIPDataSet.cxx, DumpToSQLITE3.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, Extract-EncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBroken-J2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakelIdentifyFile.cxx, GenFakelImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, PublicDict.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLength-SQIVR.cxx, rle2img.cxx, SimpleScanner.cxx, SortImage.cxx, StreamImageReaderTest.cxx, TraverseModules.cxx, and VolumeSorter.cxx.

25.258.2 Constructor & Destructor Documentation

25.258.2.1 `gdcm::Tag::Tag (uint16_t group, uint16_t element)` `[inline]`

Constructor with 2*uint16_t.

25.258.2.2 `gdcm::Tag::Tag (uint32_t tag = 0)` `[inline]`

Constructor with 1*uint32_t Prefer the ctor that takes two uint16_t.

25.258.2.3 `gdcm::Tag::Tag (const Tag & val)` `[inline]`

References tag.

25.258.3 Member Function Documentation

25.258.3.1 `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()`, and `SetPrivate-Creator()`.

25.258.3.2 `uint32_t gdcm::Tag::GetElementTag () const` `[inline]`

Returns the full tag value of the given Tag.

25.258.3.3 `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::FileMeta-Information::Insert()`, `gdcM::DataSet::Insert()`, `IsGroupXX()`, `gdcM::PrivateDict::PrintXML()`, `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()`.

25.258.3.4 `uint32_t gdcM::Tag::GetLength () const [inline]`

return the length of tag (read: size on disk)

25.258.3.5 `Tag gdcM::Tag::GetPrivateCreator () const [inline]`

Return the Private Creator Data Element tag of a private data element.

References `SetElement()`.

25.258.3.6 `bool gdcM::Tag::IsGroupLength () const [inline]`

return whether the tag correspond to a group length tag:

25.258.3.7 `bool gdcM::Tag::IsGroupXX (const Tag & t) const [inline]`

e.g 6002,3000 belong to groupXX: 6000,3000

References `GetElement()`, `GetGroup()`, and `IsPrivate()`.

25.258.3.8 `bool gdcM::Tag::IsIllegal () const [inline]`

return if the tag is considered to be an illegal tag

25.258.3.9 `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()`.

25.258.3.10 `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.

25.258.3.11 `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...

25.258.3.12 `bool gdcmm::Tag::operator!= (const Tag & _val) const [inline]`

References tag.

25.258.3.13 `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.

25.258.3.14 `bool gdcmm::Tag::operator<= (const Tag & t2) const [inline]`

25.258.3.15 `Tag& gdcmm::Tag::operator= (const Tag & _val) [inline]`

References tag.

25.258.3.16 `bool gdcmm::Tag::operator== (const Tag & _val) const [inline]`

References tag.

25.258.3.17 `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)

25.258.3.18 `uint16_t& gdcmm::Tag::operator[] (const unsigned int & _id) [inline]`

Returns the Group or Element of the given Tag, depending on id (0/1)

25.258.3.19 `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`

25.258.3.20 `template<typename TSwap> std::istream& gdcm::Tag::Read (std::istream & is)` `[inline]`

Read a tag from binary representation.

25.258.3.21 `bool gdcm::Tag::ReadFromCommaSeparatedString (const char * str)`

Read from a comma separated string. This is a highly user oriented function, the string should be formatted as: 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

Reimplemented in `gdcm::PrivateTag`.

25.258.3.22 `bool gdcm::Tag::ReadFromPipeSeparatedString (const char * str)`

Read from a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

`ReadFromCommaSeparatedString`

25.258.3.23 `void gdcm::Tag::SetElement (uint16_t element)` `[inline]`

Sets the 'Element number' of the given Tag.

Examples:

`DuplicatePCDE.cxx`, and `PublicDict.cxx`.

Referenced by `GetPrivateCreator()`, and `gdcm::operator>>()`.

25.258.3.24 `void gdcm::Tag::SetElementTag (uint16_t group, uint16_t element)` `[inline]`

Sets the 'Group number' & 'Element number' of the given Tag.

25.258.3.25 `void gdcm::Tag::SetElementTag (uint32_t tag)` `[inline]`

Sets the full tag value of the given Tag.

25.258.3.26 `void gdcm::Tag::SetGroup (uint16_t group)` `[inline]`

Sets the 'Group number' of the given Tag.

Referenced by `gdcm::operator>>()`.

25.258.3.27 `void gdcm::Tag::SetPrivateCreator (Tag const & t) [inline]`

Set private creator:

Examples:

DuplicatePCDE.cxx.

References `GetElement()`, and `IsPrivate()`.

25.258.3.28 `template<typename TSwap > const std::ostream& gdcm::Tag::Write (std::ostream & os) const [inline]`

Write a tag in binary rep.

Referenced by `gdcm::SequenceOfFragments::Write()`, `gdcm::SequenceOfItems::Write()`, and `gdcm::Item::Write()`.

25.258.4 Friends And Related Function Documentation

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

25.258.4.2 `std::istream& operator>> (std::istream & _is, Tag & _val) [friend]`

25.258.5 Member Data Documentation

25.258.5.1 `char gdcm::Tag::bytes[4]`

25.258.5.2 `uint32_t gdcm::Tag::tag`

Referenced by `operator!=()`, `operator<()`, `operator=()`, `operator==()`, and `Tag()`.

25.258.5.3 `uint16_t gdcm::Tag::tags[2]`

Referenced by `operator<()`.

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

- `gdcmTag.h`

25.259 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.

25.259.1 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

25.259.2 Constructor & Destructor Documentation

25.259.2.1 gdcmm::TagPath::TagPath ()

25.259.2.2 gdcmm::TagPath::~~TagPath ()

25.259.3 Member Function Documentation

25.259.3.1 bool gdcmm::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

25.259.3.2 bool gdcmm::TagPath::ConstructFromTagList (Tag const * *l*, unsigned int *n*)

Construct from a list of tags.

25.259.3.3 static bool gdcmm::TagPath::IsValid (const char * *path*) [static]

Return if path is valid or not.

25.259.3.4 void gdcmm::TagPath::Print (std::ostream &) const

25.259.3.5 bool gdcmm::TagPath::Push (Tag const & *t*)

25.259.3.6 bool gdcmm::TagPath::Push (unsigned int *itemnum*)

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

- gdcmmTagPath.h

25.260 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 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.

25.260.1 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

25.260.2 Member Typedef Documentation

25.260.2.1 `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

25.260.2.2 `typedef const char* const(* gdcm::Testing::MediaStorageDataFileType)[2]`

return the table that map the media storage (as string) of a filename (gdcmData)

25.260.3 Constructor & Destructor Documentation

25.260.3.1 `gdcm::Testing::Testing () [inline]`

25.260.3.2 `gdcm::Testing::~~Testing () [inline]`

25.260.4 Member Function Documentation

25.260.4.1 `static bool gdcm::Testing::ComputeFileMD5 (const char * filename, char digest_str[33]) [static]`

25.260.4.2 `static bool gdcm::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 gdcm::MD5 API when doing md5 computation.

25.260.4.3 `static const char* gdcm::Testing::GetDataExtraRoot () [static]`

Return the GDCM DATA EXTRA ROOT.

Examples:

DiscriminateVolume.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

25.260.4.4 `static const char* gdcm::Testing::GetDataRoot () [static]`

Return the GDCM DATA ROOT.

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, and Magnify-File.cxx.

25.260.4.5 `static const char* gdcm::Testing::GetFileName (unsigned int file) [static]`

25.260.4.6 `static const char* const* gdcm::Testing::GetFileNames () [static]`

return the table of fullpath to gdcmData DICOM files:

Examples:

TestReader.cxx.

25.260.4.7 `static int gdcm::Testing::GetLossyFlagFromFile (const char * filepath) [static]`

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

25.260.4.8 `static const char* const* gdcm::Testing::GetMD5DataImage (unsigned int file) [static]`

25.260.4.9 `static MD5DataImagesType gdcm::Testing::GetMD5DataImages () [static]`

25.260.4.10 `static const char* gdcm::Testing::GetMD5FromBrokenFile (const char * filepath) [static]`

Return what should have been the md5 of file 'filepath' This is based on current GDCM implementation to decipher a broken DICOM file.

25.260.4.11 `static const char* gdcm::Testing::GetMD5FromFile (const char * filepath) [static]`

25.260.4.12 `static const char* const* gdcm::Testing::GetMediaStorageDataFile (unsigned int file) [static]`

25.260.4.13 `static MediaStorageDataFilesType gdcm::Testing::GetMediaStorageDataFiles () [static]`

25.260.4.14 `static const char* gdcm::Testing::GetMediaStorageFromFile (const char * filepath) [static]`

Examples:

TestReader.cxx.

25.260.4.15 `static unsigned int gdcm::Testing::GetNumberOfFileNames () [static]`

25.260.4.16 `static unsigned int gdcm::Testing::GetNumberOfMD5DataImages () [static]`

25.260.4.17 `static unsigned int gdcmm::Testing::GetNumberOfMediaStorageDataFiles () [static]`

25.260.4.18 `static const char* gdcmm::Testing::GetPixelSpacingDataRoot () [static]`

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

25.260.4.19 `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

25.260.4.20 `static const char* gdcmm::Testing::GetSourceDirectory () [static]`

25.260.4.21 `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

25.260.4.22 `static const char* gdcmm::Testing::GetTempDirectory (const char * subdir = 0) [static]`

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

25.260.4.23 `static const wchar_t* gdcmm::Testing::GetTempDirectoryW (const wchar_t * subdir = 0) [static]`

NOT THREAD SAFE.

25.260.4.24 `static const char* gdcmm::Testing::GetTempFilename (const char * filename, const char * subdir = 0) [static]`

NOT THREAD SAFE.

25.260.4.25 `static const wchar_t* gdcmm::Testing::GetTempFilenameW (const wchar_t * filename, const wchar_t * subdir = 0) [static]`

NOT THREAD SAFE.

25.260.4.26 `void gdcmm::Testing::Print (std::ostream & os = std::cout)`

Print.

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

- `gdcmmTesting.h`

25.261 gdcmm::Trace Class Reference

Trace.

```
#include <gdcmmTrace.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 bool GetErrorFlag ()`
- `static std::ostream & GetStream ()`
- `static bool GetWarningFlag ()`
- `static void SetDebug (bool debug)`
- `static void SetError (bool debug)`
- `static void SetStream (std::ostream &os)`
Explicitely set the ostream for gdcm::Trace to report to.
- `static void SetWarning (bool debug)`
- `static void WarningOff ()`
- `static void WarningOn ()`

25.261.1 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.

25.261.2 Constructor & Destructor Documentation

25.261.2.1 `gdcm::Trace::Trace ()`

25.261.2.2 `gdcm::Trace::~~Trace ()`

25.261.3 Member Function Documentation

25.261.3.1 `static void gdcm::Trace::DebugOff ()` `[static]`

Examples:

TestReader.cxx.

25.261.3.2 `static void gdcm::Trace::DebugOn ()` `[static]`

Examples:

Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

```

25.261.3.3 static void gdcm::Trace::ErrorOff ( ) [static]
25.261.3.4 static void gdcm::Trace::ErrorOn ( ) [static]
25.261.3.5 static bool gdcm::Trace::GetDebugFlag ( ) [static]
25.261.3.6 static bool gdcm::Trace::GetErrorFlag ( ) [static]
25.261.3.7 static std::ostream& gdcm::Trace::GetStream ( ) [static]
25.261.3.8 static bool gdcm::Trace::GetWarningFlag ( ) [static]
25.261.3.9 static void gdcm::Trace::SetDebug ( bool debug ) [static]

```

Examples:

DumpToSQLITE3.cxx.

```

25.261.3.10 static void gdcm::Trace::SetError ( bool debug ) [static]
25.261.3.11 static void gdcm::Trace::SetStream ( std::ostream & os ) [static]

```

Explicitely set the ostream for gdcm::Trace to report to.

```

25.261.3.12 static void gdcm::Trace::SetWarning ( bool debug ) [static]

```

Examples:

DumpToSQLITE3.cxx.

```

25.261.3.13 static void gdcm::Trace::WarningOff ( ) [static]

```

Examples:

TestReader.cxx.

```

25.261.3.14 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

25.262 gdcm::TransferSyntax Class Reference

Class to manipulate Transfer Syntax.

```
#include <gdcmTransferSyntax.h>
```

Public Types

- enum NegotiatedType {
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,
JPIPReferenced,
RLELossless,
MPEG2MainProfile,
ImplicitVRBigEndianACRNEMA,
CT_private_ELE,
TS_END }

Public Member Functions

- TransferSyntax (TSType type=ImplicitVRLittleEndian)
- bool CanStoreLossy () const
- NegotiatedType GetNegotiatedType () 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)`

25.262.1 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`, and `LargeVRDSExplicit.cxx`.

25.262.2 Member Enumeration Documentation

25.262.2.1 enum `gdcm::TransferSyntax::NegociatedType`

Enumerator:

Unknown

Explicit

Implicit

25.262.2.2 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**JPIPRendered**RLELossless**MPEG2MainProfile**ImplicitVRBigEndianACRNEMA**CT_private_ELE**TS_END*

25.262.3 Constructor & Destructor Documentation

25.262.3.1 `gdcm::TransferSyntax::TransferSyntax (TSType type = ImplicitVRLittleEndian)` `[inline]`

25.262.4 Member Function Documentation

25.262.4.1 `bool gdcm::TransferSyntax::CanStoreLossy () const`

return if TransFer Syntax Allow storing of Lossy Pixel Data

25.262.4.2 `NegotiatedType gdcm::TransferSyntax::GetNegociatedType () const`

25.262.4.3 `const char* gdcm::TransferSyntax::GetString () const` `[inline]`

References GetTSString().

25.262.4.4 `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.

25.262.4.5 `static const char* gdcm::TransferSyntax::GetTSString (TSType ts)` `[static]`

Examples:

LargeVRDSExplicit.cxx.

Referenced by GetString(), and gdcm::operator<<().

25.262.4.6 `static TSType gdcM::TransferSyntax::GetTSType (const char * str)` `[static]`

25.262.4.7 `bool gdcM::TransferSyntax::IsEncapsulated () const`

Examples:

ExtractIconFromFile.cxx.

25.262.4.8 `bool gdcM::TransferSyntax::IsEncoded () const`

25.262.4.9 `bool gdcM::TransferSyntax::IsExplicit () const`

25.262.4.10 `bool gdcM::TransferSyntax::IsImplicit () const`

25.262.4.11 `bool gdcM::TransferSyntax::IsLossless () const`

Return if the transfer syntax algorithm is a lossless algorithm

25.262.4.12 `bool gdcM::TransferSyntax::IsLossy () const`

Return if the transfer syntax algorithm is a lossy algorithm

25.262.4.13 `bool gdcM::TransferSyntax::IsValid () const` `[inline]`

25.262.4.14 `gdcM::TransferSyntax::operator TSType () const` `[inline]`

25.262.5 Friends And Related Function Documentation

25.262.5.1 `std::ostream& operator<< (std::ostream & os, const TransferSyntax & ts)` `[friend]`

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

- gdcMTransferSyntax.h

25.263 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

25.263.1 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

25.263.2 Constructor & Destructor Documentation

25.263.2.1 gdcm::network::TransferSyntaxSub::TransferSyntaxSub ()

25.263.3 Member Function Documentation

25.263.3.1 const char* gdcm::network::TransferSyntaxSub::GetName () const [inline]

25.263.3.2 bool gdcm::network::TransferSyntaxSub::operator== (const TransferSyntaxSub & ts) const [inline]

25.263.3.3 void gdcm::network::TransferSyntaxSub::Print (std::ostream & os) const

25.263.3.4 std::istream& gdcm::network::TransferSyntaxSub::Read (std::istream & is)

25.263.3.5 void gdcm::network::TransferSyntaxSub::SetName (const char * name)

25.263.3.6 void gdcm::network::TransferSyntaxSub::SetNameFromUID (UIDs::TSName tsname)

25.263.3.7 size_t gdcm::network::TransferSyntaxSub::Size () const

25.263.3.8 const std::ostream& gdcm::network::TransferSyntaxSub::Write (std::ostream & os) const

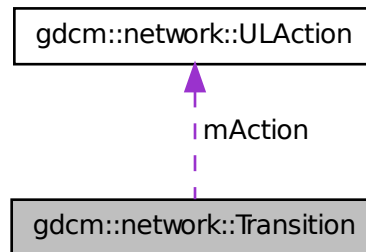
The documentation for this class was generated from the following file:

- gdcmTransferSyntaxSub.h

25.264 gdcm::network::Transition Struct Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for `gdcmm::network::Transition`:



Public Member Functions

- `Transition ()`
- `Transition (int inEndState, ULAction *inAction)`
- `~Transition ()`

Static Public Member Functions

- `static Transition * MakeNew (int inEndState, ULAction *inAction)`

Public Attributes

- `ULAction * mAction`
- `int mEnd`

25.264.1 Constructor & Destructor Documentation

25.264.1.1 `gdcmm::network::Transition::Transition ()` `[inline]`

References `gdcmm::network::eStaDoesNotExist`, `mAction`, and `mEnd`.

Referenced by `MakeNew()`.

25.264.1.2 `gdcmm::network::Transition::~~Transition ()` `[inline]`

References `mAction`.

25.264.1.3 `gdcmm::network::Transition::Transition (int inEndState, ULAction * inAction)` `[inline]`

References `mAction`, and `mEnd`.

25.264.2 Member Function Documentation

25.264.2.1 `static Transition* gdcm::network::Transition::MakeNew (int inEndState, ULAction * inAction) [inline], [static]`

References `Transition()`.

25.264.3 Member Data Documentation

25.264.3.1 `ULAction* gdcm::network::Transition::mAction`

Referenced by `Transition()`, and `~Transition()`.

25.264.3.2 `int gdcm::network::Transition::mEnd`

Referenced by `Transition()`.

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

- `gdcmULTransitionTable.h`

25.265 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)`

25.265.1 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.

25.265.2 Member Enumeration Documentation

25.265.2.1 enum `gdcmm::Type::TypeType`

Enumerator:

T1
T1C
T2
T2C
T3
UNKNOWN

25.265.3 Constructor & Destructor Documentation

25.265.3.1 `gdcmm::Type::Type (TypeType type = UNKNOWN) [inline]`

25.265.4 Member Function Documentation

25.265.4.1 `static const char* gdcmm::Type::GetTypeString (TypeType type) [static]`

Referenced by `gdcmm::operator<<()`.

25.265.4.2 `static TypeType gdcmm::Type::GetTypeType (const char * type) [static]`

Referenced by `gdcmm::ModuleEntry::ModuleEntry()`.

25.265.4.3 `gdcm::Type::operator TypeType () const` `[inline]`

25.265.5 Friends And Related Function Documentation

25.265.5.1 `std::ostream& operator<< (std::ostream & os, const Type & vr)` `[friend]`

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

- `gdcmType.h`

25.266 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

Public Attributes

- `char Internal [64+1]`

Friends

- `std::ostream & operator<< (std::ostream &_os, const UI &_val)`

25.266.1 Friends And Related Function Documentation

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

25.266.2 Member Data Documentation

25.266.2.1 `char gdcm::UI::Internal[64+1]`

Referenced by `gdcm::operator<<()`.

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

- `gdcmVR.h`

25.267 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)

25.267.1 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.

25.267.2 Constructor & Destructor Documentation

25.267.2.1 gdcm::UIDGenerator::UIDGenerator () [inline]

By default the root of a UID is a GDCM Root...

25.267.3 Member Function Documentation

25.267.3.1 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, StreamImageReaderTest.cxx, and uid_unique.cxx.

25.267.3.2 static bool gdcm::UIDGenerator::GenerateUUID (unsigned char * uuid_data) [static],[protected]

25.267.3.3 static const char* gdcm::UIDGenerator::GetGDCMUID () [static]

Return the default (GDCM) root UID:

25.267.3.4 `static const char* gdcm::UIDGenerator::GetRoot () [static]`

25.267.3.5 `static bool gdcm::UIDGenerator::IsValid (const char * uid) [static]`

Find out if the string is a valid UID or not

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

25.267.3.6 `static void gdcm::UIDGenerator::SetRoot (const char * root) [static]`

The current implementation in GDCM make use of the UUID implementation (RFC 4122) and has been successfully been tested for a root of size 26 bytes. Any longer root should work (the `::Generate()` function will return a string), but will truncate the high bits of the 128bits UUID until the generated string fits on 64 bits. The authors disclaims any responsibility for guaranteeing uniqueness of UIDs when the root is longer than 26 bytes.

Examples:

`uid_unique.cxx.`

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

- `gdcmUIDGenerator.h`

25.268 gdcm::UIDs Class Reference

all known uids

```
#include <gdcmUIDs.h>
```

Public Types

- `typedef const char *const (* TransferSyntaxStringsType)[2]`
- `enum TSName {`

```
VerificationSOPClass = 1,  
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM = 2,  
ExplicitVRLittleEndian = 3,  
DeflatedExplicitVRLittleEndian = 4,  
ExplicitVRBigEndian = 5,  
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression = 6,  
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only = 7,  
JPEGExtendedProcess35Retired = 8,  
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,
JPIPIReferenced = 30,
JPIPIReferencedDeflate = 31,
MPEG2MainProfileMainLevel = 32,
RLELossless = 33,
RFC2557MIMEencapsulation = 34,
XMLEncoding = 35,
MediaStorageDirectoryStorage = 36,
TalairachBrainAtlasFrameofReference = 37,
SPM2T1FrameofReference = 38,
SPM2T2FrameofReference = 39,
SPM2PDFrameofReference = 40,
SPM2EPIFrameofReference = 41,
SPM2FILT1FrameofReference = 42,
SPM2PETFrameofReference = 43,
SPM2TRANSMFrameofReference = 44,
SPM2SPECTFrameofReference = 45,
SPM2GRAYFrameofReference = 46,
SPM2WHITEFrameofReference = 47,
SPM2CSFFFrameofReference = 48,
SPM2BRAINMASKFrameofReference = 49,
SPM2AVG305T1FrameofReference = 50,
SPM2AVG152T1FrameofReference = 51,
SPM2AVG152T2FrameofReference = 52,
SPM2AVG152PDFrameofReference = 53,
SPM2SINGLESUBJT1FrameofReference = 54,
ICBM452T1FrameofReference = 55,
ICBMSingleSubjectMRIFrameofReference = 56,
BasicStudyContentNotificationSOPClassRetired = 57,
StorageCommitmentPushModelSOPClass = 58,
StorageCommitmentPushModelSOPInstance = 59,
StorageCommitmentPullModelSOPClassRetired = 60,
StorageCommitmentPullModelSOPInstanceRetired = 61,
ProceduralEventLoggingSOPClass = 62,
ProceduralEventLoggingSOPInstance = 63,
SubstanceAdministrationLoggingSOPClass = 64,
SubstanceAdministrationLoggingSOPInstance = 65,
DICOMUIDRegistry = 66,
DICOMControlledTerminology = 67,
DICOMApplicationContextName = 68,
DetachedPatientManagementSOPClassRetired = 69,
DetachedPatientManagementMetaSOPClassRetired = 70,
DetachedVisitManagementSOPClassRetired = 71,
DetachedStudyManagementSOPClassRetired = 72,
StudyComponentManagementSOPClassRetired = 73,
ModalityPerformedProcedureStepSOPClass = 74,
ModalityPerformedProcedureStepRetrieveSOPClass = 75,
ModalityPerformedProcedureStepNotificationSOPClass = 76,
DetachedResultsManagementSOPClassRetired = 77,
DetachedResultsManagementMetaSOPClassRetired = 78,
DetachedStudyManagementMetaSOPClassRetired = 79,
DetachedInterpretationManagementSOPClassRetired = 80,
StorageServiceClass = 81,
BasicFilmSessionSOPClass = 82,

VLWholeSlideMicroscopyImageStorage }

• 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_5_1_4_1_1_77_1_6 }
```

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)

25.268.1 Detailed Description

all known uids

Examples:

```
GenerateStandardSOPClasses.cxx.
```

25.268.2 Member Typedef Documentation

25.268.2.1 `typedef const char* const(* gdcmm::UIDs::TransferSyntaxStringsType)[2]`

25.268.3 Member Enumeration Documentation

25.268.3.1 `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

25.268.3.2 enum gdcM::UIDs::TSType

Enumerator:

uid_1_2_840_10008_1_1
uid_1_2_840_10008_1_2
uid_1_2_840_10008_1_2_1
uid_1_2_840_10008_1_2_1_99
uid_1_2_840_10008_1_2_2
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

25.268.4 Member Function Documentation

25.268.4.1 `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<<()`.

25.268.4.2 `static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings () [static]`

25.268.4.3 `const char* gdcm::UIDs::GetString () const`

When object is Initialize function return the uid return NULL when not initialized

Examples:

GenerateStandardSOPClasses.cxx.

Referenced by `gdcm::operator<<()`.

25.268.4.4 `static const char* const* gdcm::UIDs::GetTransferSyntaxString (unsigned int ts) [static]`

25.268.4.5 `static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings () [static]`

25.268.4.6 `static const char* gdcm::UIDs::GetUIDName (unsigned int ts) [static]`

25.268.4.7 `static const char* gdcm::UIDs::GetUIDString (unsigned int ts) [static]`

25.268.4.8 `gdcm::UIDs::operator TSType () const [inline]`

25.268.4.9 `bool gdcm::UIDs::SetFromUID (const char * str)`

Initialize object from a string (a uid number) return false on error, and internal state is set to 0

Examples:

GenerateStandardSOPClasses.cxx.

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

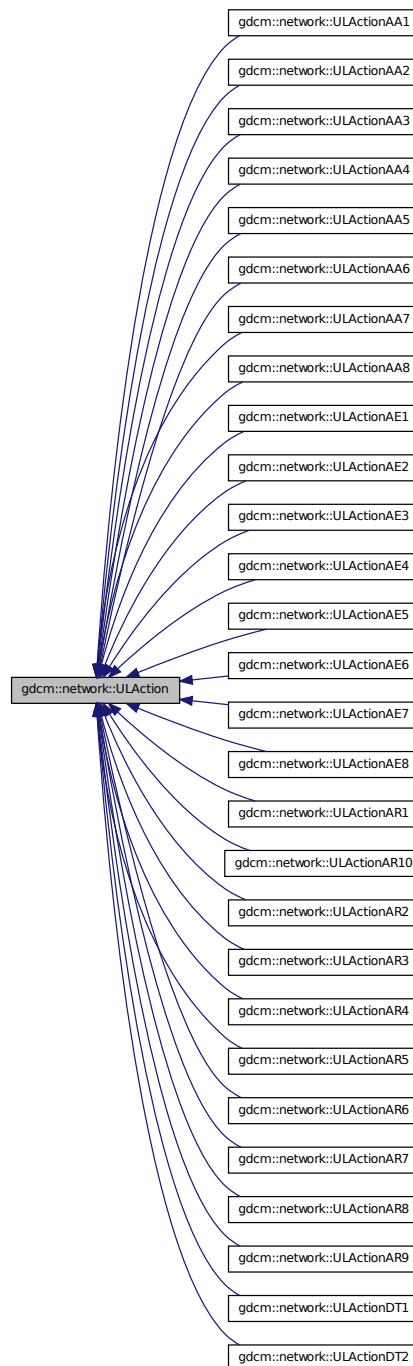
- `gdcmUIDs.h`

25.269 gdcm::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 <gdcmULAction.h>
```

Inheritance diagram for gdcmm::network::ULAction:



Public Member Functions

- `ULAction ()`
- `virtual ~ULAction ()`

- virtual EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)=0

25.269.1 Detailed Description

ULAction A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

Essentially, the ULConnectionManager will take this object, determined from the current ULState of the ULConnection, and pass the ULConnection object to the ULAction. The ULAction will then invoke whatever necessary commands are required by a given action.

The result of a ULAction is a ULEvent (ie, what happened as a result of the action).

This ULEvent is passed to the ULState, so that the transition to the next state can occur.

Actions are associated with Payloads— be thos filestreams, AETitles to establish connections, whatever. The actual parameters that the user will pass via an action will come through a Payload object, which should, in itself, be some gdcmm-based object (but not all objects can be payloads; sending a single dataelement as a payload isn't meaningful). As such, each action has its own particular payload.

For the sake of keeping files together, both the particular payload class and the action class will be defined in the same header file. Payloads should JUST be data (or streams), NO METHODS.

Some actions perform changes that should raise events on the local system, and some actions perform changes that will require waiting for events from the remote system.

Therefore, this base action has been modified so that those events are set by each action. When the event loop runs an action, it will then test to see if a local event was raised by the action, and if so, perform the appropriate subsequent action. If the action requires waiting for a response from the remote system, then the event loop will sit there (presumably with the ARTIM timer running) and wait for a response from the remote system. Once a response is obtained, then the the rest of the state transitions can happen.

25.269.2 Constructor & Destructor Documentation

25.269.2.1 gdcmm::network::ULAction::ULAction () [inline]

25.269.2.2 virtual gdcmm::network::ULAction::~~ULAction () [inline], [virtual]

25.269.3 Member Function Documentation

25.269.3.1 virtual EStateID gdcmm::network::ULAction::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [pure virtual]

Implemented in gdcmm::network::ULActionAR10, gdcmm::network::ULActionAR9, gdcmm::network::ULActionAE8, gdcmm::network::ULActionAA8, gdcmm::network::ULActionAR8, gdcmm::network::ULActionAE7, gdcmm::network::ULActionAA7, gdcmm::network::ULActionAR7, gdcmm::network::ULActionAE6, gdcmm::network::ULActionAA6, gdcmm::network::ULActionAR6, gdcmm::network::ULActionAA5, gdcmm::network::ULActionAE5, gdcmm::network::ULActionAR5, gdcmm::network::ULActionAA4, gdcmm::network::ULActionAE4, gdcmm::network::ULActionAR4, gdcmm::network::ULActionAA3, gdcmm::network::ULActionAE3, gdcmm::network::ULActionAR3, gdcmm::network::ULActionAA2, gdcmm::network::ULActionAE2, gdcmm::network::ULActionAR2, gdcmm::network::ULActionDT2, gdcmm::network::ULActionAA1, gdcmm::network::ULActionAE1, gdcmm::network::ULActionAR1, and gdcmm::network::ULActionDT1.

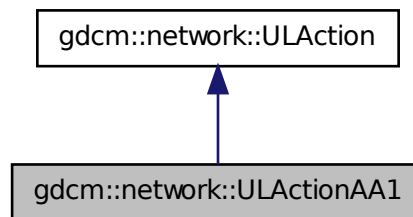
The documentation for this class was generated from the following file:

- gdcmmULAction.h

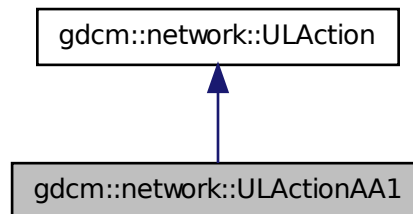
25.270 gdcm::network::ULActionAA1 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA1:



Collaboration diagram for gdcm::network::ULActionAA1:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.270.1 Member Function Documentation

25.270.1.1 `EStateID gdcm::network::ULActionAA1::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` `[virtual]`

Implements `gdcm::network::ULAction`.

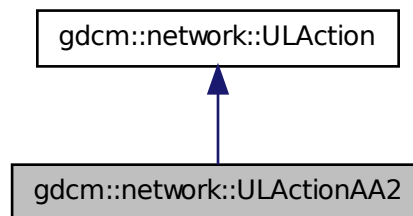
The documentation for this class was generated from the following file:

- gdcmlActionAA.h

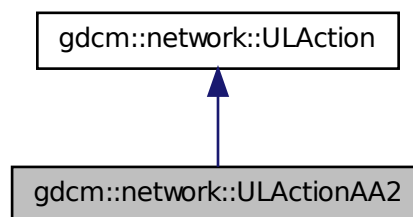
25.271 gdcmlnetwork::ULActionAA2 Class Reference

```
#include <gdcmlActionAA.h>
```

Inheritance diagram for gdcmlnetwork::ULActionAA2:



Collaboration diagram for gdcmlnetwork::ULActionAA2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.271.1 Member Function Documentation

25.271.1.1 **EStateID** gdcm::network::ULActionAA2::PerformAction (**Subject** * *s*, **ULError** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

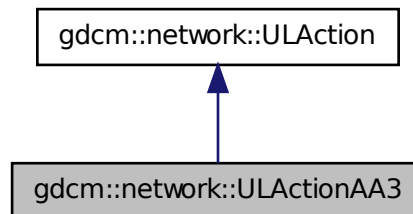
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

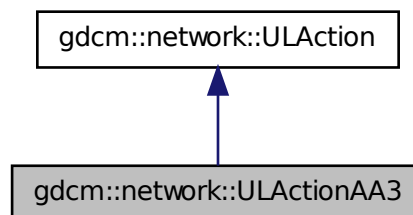
25.272 gdcm::network::ULActionAA3 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA3:



Collaboration diagram for gdcm::network::ULActionAA3:



Public Member Functions

- **EStateID** PerformAction (Subject *s, ULError &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.272.1 Member Function Documentation

25.272.1.1 **EStateID** `gdcmm::network::ULActionAA3::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements `gdcmm::network::ULAction`.

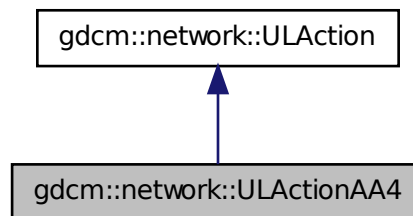
The documentation for this class was generated from the following file:

- `gdcmmULActionAA.h`

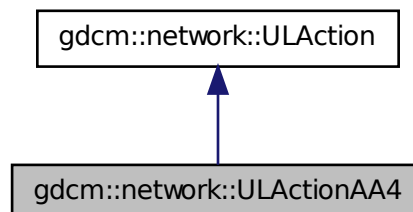
25.273 `gdcmm::network::ULActionAA4` Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for `gdcmm::network::ULActionAA4`:



Collaboration diagram for `gdcmm::network::ULActionAA4`:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.273.1 Member Function Documentation

25.273.1.1 EStateID gdcm::network::ULActionAA4::PerformAction (Subject * s, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

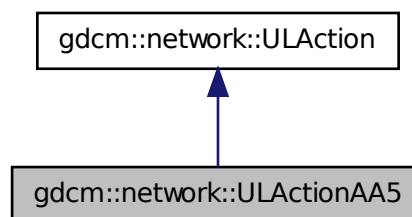
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

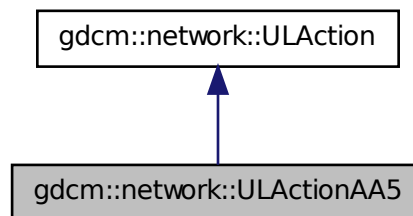
25.274 gdcm::network::ULActionAA5 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA5:



Collaboration diagram for `gdcm::network::ULActionAA5`:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.274.1 Member Function Documentation

25.274.1.1 `EStateID gdcm::network::ULActionAA5::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcm::network::ULAction`.

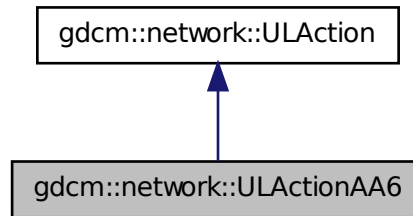
The documentation for this class was generated from the following file:

- `gdcmULActionAA.h`

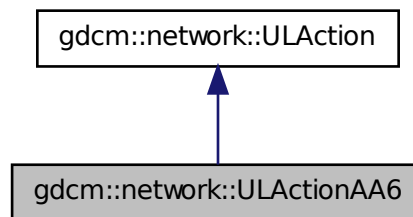
25.275 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA6:



Collaboration diagram for gdcmm::network::ULActionAA6:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.275.1 Member Function Documentation

25.275.1.1 `EStateID gdcmm::network::ULActionAA6::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcmm::network::ULAction`.

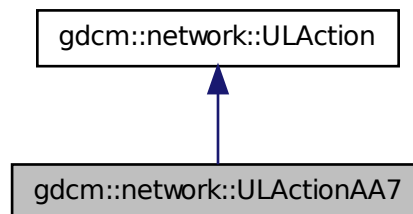
The documentation for this class was generated from the following file:

- `gdcmmULActionAA.h`

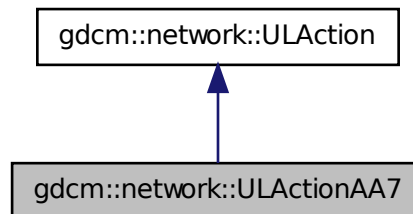
25.276 gdcmm::network::ULActionAA7 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA7:



Collaboration diagram for gdcmm::network::ULActionAA7:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.276.1 Member Function Documentation

25.276.1.1 `EStateID gdcmm::network::ULActionAA7::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` `[virtual]`

Implements `gdcmm::network::ULAction`.

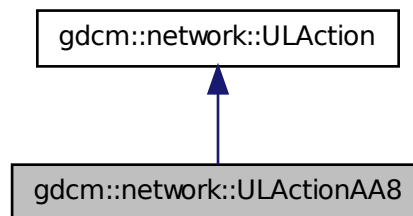
The documentation for this class was generated from the following file:

- gdcmmULActionAA.h

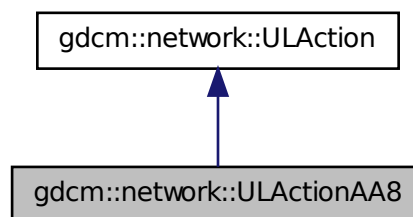
25.277 gdcmm::network::ULActionAA8 Class Reference

```
#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 &outWaitingForEvent, EEventID &outRaisedEvent)

25.277.1 Member Function Documentation

25.277.1.1 **EStateID** `gdcmm::network::ULActionAA8::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcmm::network::ULAction`.

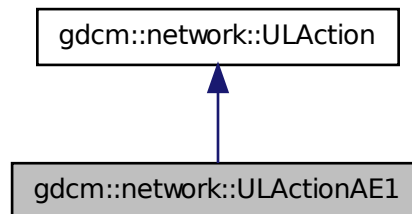
The documentation for this class was generated from the following file:

- `gdcmmULActionAA.h`

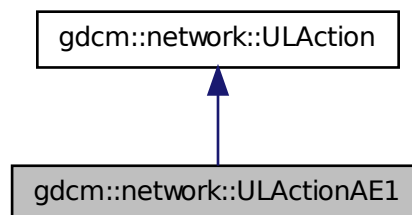
25.278 `gdcmm::network::ULActionAE1` Class Reference

```
#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 &outWaitingForEvent, EEventID &outRaisedEvent)`

25.278.1 Member Function Documentation

25.278.1.1 **EStateID** gdcn::network::ULActionAE1::PerformAction (**Subject** * *s*, **UEvent** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements gdcn::network::ULAction.

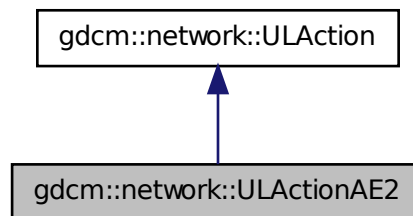
The documentation for this class was generated from the following file:

- gdcnULActionAE.h

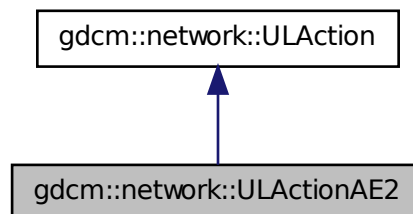
25.279 gdcn::network::ULActionAE2 Class Reference

```
#include <gdcnULActionAE.h>
```

Inheritance diagram for gdcn::network::ULActionAE2:



Collaboration diagram for gdcn::network::ULActionAE2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.279.1 Member Function Documentation

25.279.1.1 EStateID gdcmm::network::ULActionAE2::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]

Implements gdcmm::network::ULAction.

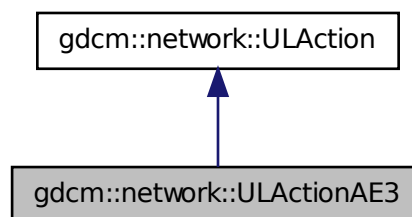
The documentation for this class was generated from the following file:

- gdcmmULActionAE.h

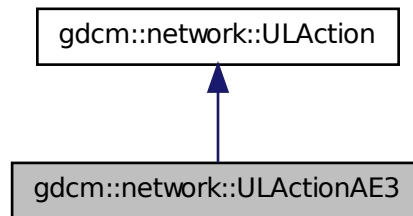
25.280 gdcmm::network::ULActionAE3 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE3:



Collaboration diagram for gdcm::network::ULActionAE3:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.280.1 Member Function Documentation

25.280.1.1 `EStateID gdcm::network::ULActionAE3::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcm::network::ULAction`.

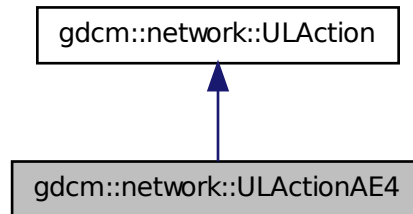
The documentation for this class was generated from the following file:

- `gdcmULActionAE.h`

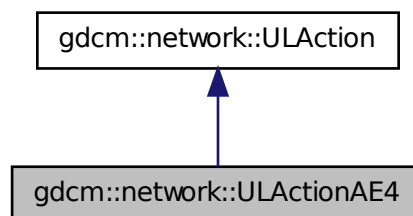
25.281 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for `gdcn::network::ULActionAE4`:



Collaboration diagram for `gdcn::network::ULActionAE4`:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.281.1 Member Function Documentation

25.281.1.1 `EStateID gdcn::network::ULActionAE4::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcn::network::ULAction`.

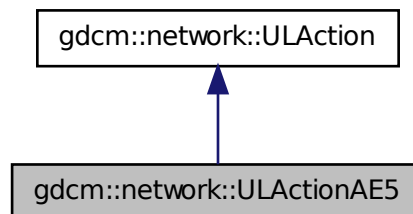
The documentation for this class was generated from the following file:

- `gdcnULActionAE.h`

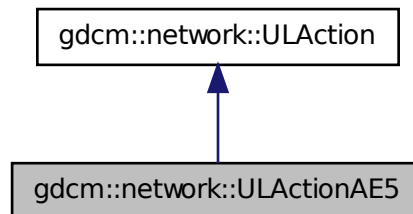
25.282 gdcm::network::ULActionAE5 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE5:



Collaboration diagram for gdcm::network::ULActionAE5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.282.1 Member Function Documentation

25.282.1.1 EStateID gdcm::network::ULActionAE5::PerformAction (Subject * s, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

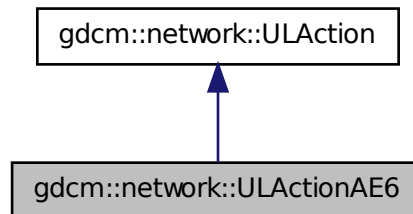
The documentation for this class was generated from the following file:

- `gdcmlActionAE.h`

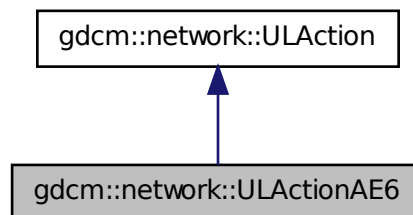
25.283 `gdcmlnetwork::ULActionAE6` Class Reference

```
#include <gdcmlActionAE.h>
```

Inheritance diagram for `gdcmlnetwork::ULActionAE6`:



Collaboration diagram for `gdcmlnetwork::ULActionAE6`:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.283.1 Member Function Documentation

25.283.1.1 **EStateID** gdcmm::network::ULActionAE6::PerformAction (**Subject** * *s*, **ULError** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements gdcmm::network::ULAction.

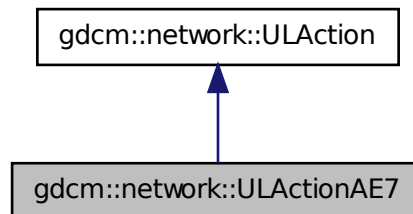
The documentation for this class was generated from the following file:

- gdcmmULActionAE.h

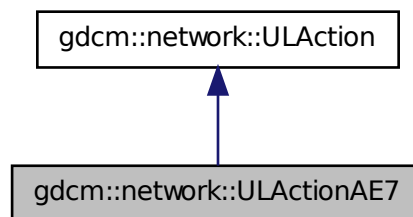
25.284 gdcmm::network::ULActionAE7 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE7:



Collaboration diagram for gdcmm::network::ULActionAE7:



Public Member Functions

- **EStateID** PerformAction (Subject *s, ULError &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.284.1 Member Function Documentation

25.284.1.1 **EStateID** `gdcmm::network::ULActionAE7::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements `gdcmm::network::ULAction`.

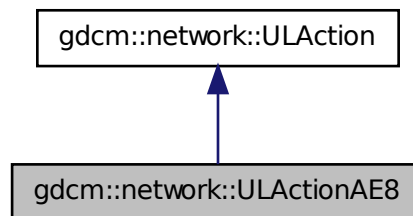
The documentation for this class was generated from the following file:

- `gdcmmULActionAE.h`

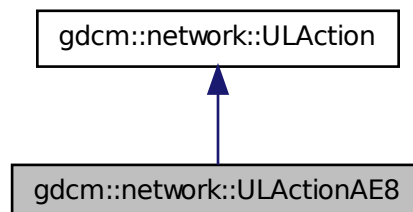
25.285 `gdcmm::network::ULActionAE8` Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for `gdcmm::network::ULActionAE8`:



Collaboration diagram for `gdcmm::network::ULActionAE8`:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.285.1 Member Function Documentation

25.285.1.1 EStateID gdcm::network::ULActionAE8::PerformAction (Subject * s, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

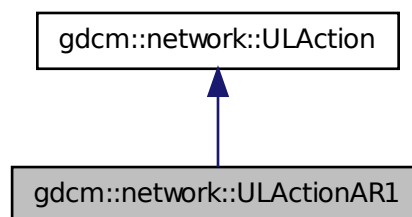
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

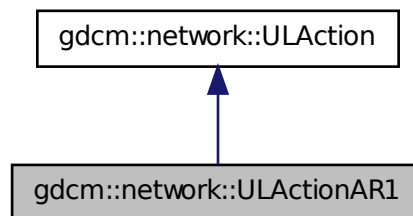
25.286 gdcm::network::ULActionAR1 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR1:



Collaboration diagram for gdcn::network::ULActionAR1:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.286.1 Member Function Documentation

25.286.1.1 `EStateID gdcn::network::ULActionAR1::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcn::network::ULAction`.

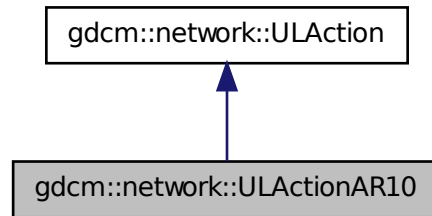
The documentation for this class was generated from the following file:

- `gdcnULActionAR.h`

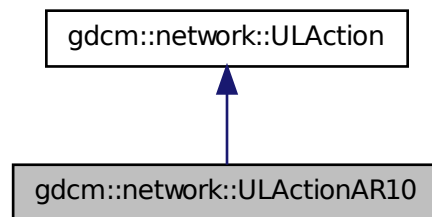
25.287 gdcn::network::ULActionAR10 Class Reference

```
#include <gdcnULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR10:



Collaboration diagram for gdcmm::network::ULActionAR10:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.287.1 Member Function Documentation

25.287.1.1 `EStateID gdcmm::network::ULActionAR10::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcmm::network::ULAction`.

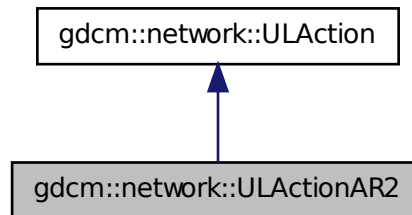
The documentation for this class was generated from the following file:

- `gdcmmULActionAR.h`

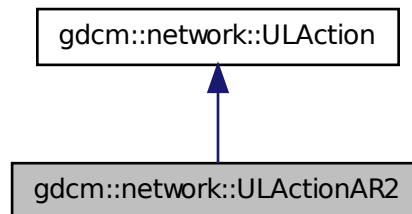
25.288 gdcmm::network::ULActionAR2 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR2:



Collaboration diagram for gdcmm::network::ULActionAR2:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.288.1 Member Function Documentation

25.288.1.1 `EStateID gdcmm::network::ULActionAR2::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` `[virtual]`

Implements `gdcmm::network::ULAction`.

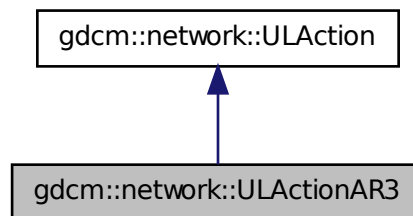
The documentation for this class was generated from the following file:

- gdcmmULActionAR.h

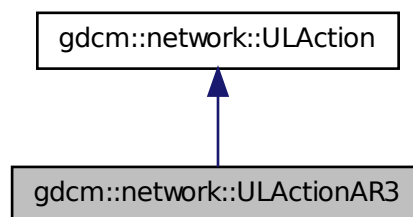
25.289 gdcmm::network::ULActionAR3 Class Reference

```
#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 &outWaitingForEvent, EEventID &outRaisedEvent)

25.289.1 Member Function Documentation

25.289.1.1 **EStateID** `gdcm::network::ULActionAR3::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements `gdcm::network::ULAction`.

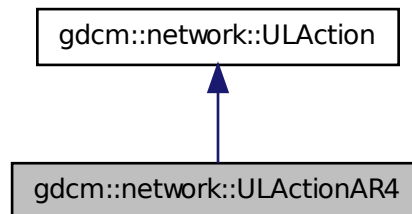
The documentation for this class was generated from the following file:

- `gdcmULActionAR.h`

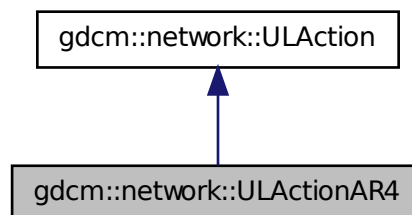
25.290 `gdcm::network::ULActionAR4` Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for `gdcm::network::ULActionAR4`:



Collaboration diagram for `gdcm::network::ULActionAR4`:



Public Member Functions

- **EStateID** `PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.290.1 Member Function Documentation

25.290.1.1 **EStateID** gdcmm::network::ULActionAR4::PerformAction (**Subject** * *s*, **UEvent** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements gdcmm::network::ULAction.

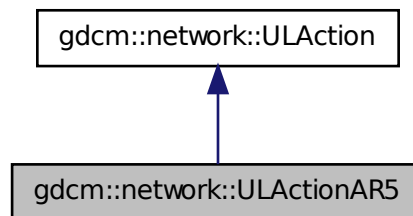
The documentation for this class was generated from the following file:

- gdcmmULActionAR.h

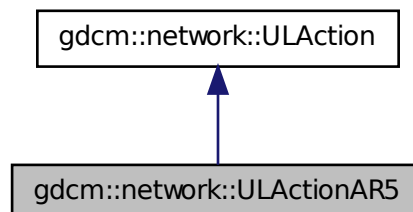
25.291 gdcmm::network::ULActionAR5 Class Reference

```
#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 &outWaitingForEvent, EEventID &outRaisedEvent)

25.291.1 Member Function Documentation

25.291.1.1 EStateID gdcmm::network::ULActionAR5::PerformAction (Subject * s, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcmm::network::ULAction.

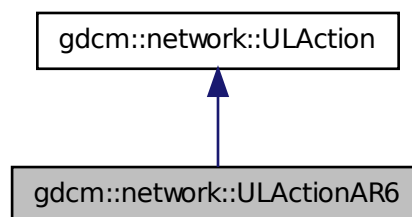
The documentation for this class was generated from the following file:

- gdcmmULActionAR.h

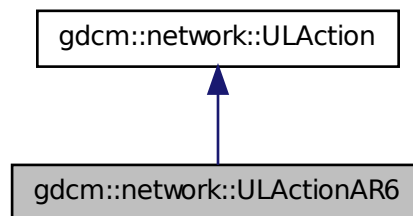
25.292 gdcmm::network::ULActionAR6 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR6:



Collaboration diagram for gdcm::network::ULActionAR6:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.292.1 Member Function Documentation

25.292.1.1 `EStateID gdcm::network::ULActionAR6::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcm::network::ULAction`.

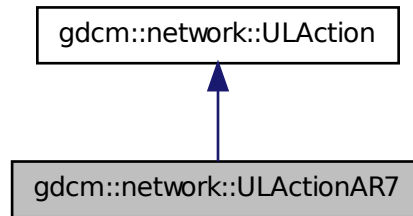
The documentation for this class was generated from the following file:

- `gdcmULActionAR.h`

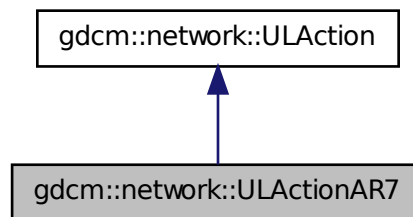
25.293 gdcm::network::ULActionAR7 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for `gdcn::network::ULActionAR7`:



Collaboration diagram for `gdcn::network::ULActionAR7`:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.293.1 Member Function Documentation

25.293.1.1 `EStateID gdcn::network::ULActionAR7::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements `gdcn::network::ULAction`.

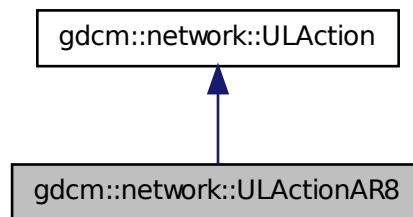
The documentation for this class was generated from the following file:

- `gdcnULActionAR.h`

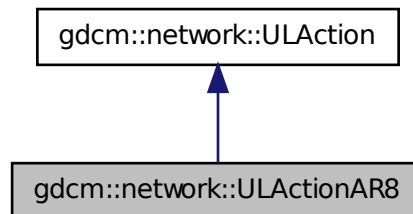
25.294 gdcm::network::ULActionAR8 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR8:



Collaboration diagram for gdcm::network::ULActionAR8:



Public Member Functions

- `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

25.294.1 Member Function Documentation

25.294.1.1 `EStateID gdcm::network::ULActionAR8::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` `[virtual]`

Implements `gdcm::network::ULAction`.

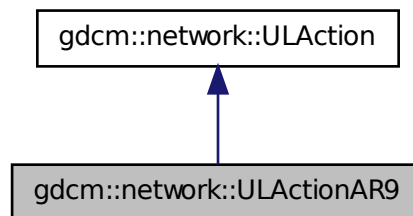
The documentation for this class was generated from the following file:

- gdcmlActionAR.h

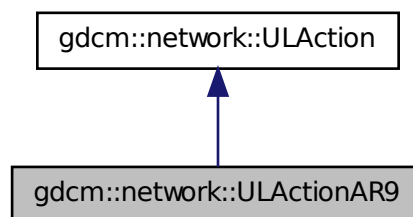
25.295 gdcml::network::ULActionAR9 Class Reference

```
#include <gdcmlActionAR.h>
```

Inheritance diagram for gdcml::network::ULActionAR9:



Collaboration diagram for gdcml::network::ULActionAR9:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.295.1 Member Function Documentation

25.295.1.1 EStateID gdcmm::network::ULActionAR9::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]

Implements gdcmm::network::ULAction.

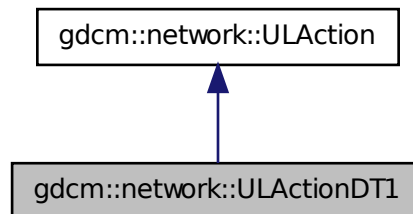
The documentation for this class was generated from the following file:

- gdcmmULActionAR.h

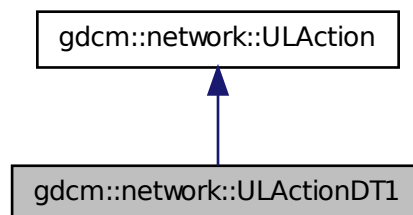
25.296 gdcmm::network::ULActionDT1 Class Reference

```
#include <gdcmmULActionDT.h>
```

Inheritance diagram for gdcmm::network::ULActionDT1:



Collaboration diagram for gdcmm::network::ULActionDT1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)

25.296.1 Member Function Documentation

25.296.1.1 **EStateID** `gdcmm::network::ULActionDT1::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements `gdcmm::network::ULAction`.

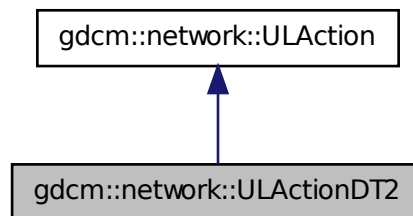
The documentation for this class was generated from the following file:

- `gdcmmULActionDT.h`

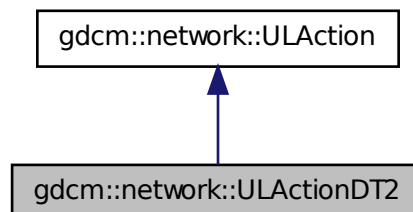
25.297 `gdcmm::network::ULActionDT2` Class Reference

```
#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 &outWaitingForEvent, EEventID &outRaisedEvent)

25.297.1 Member Function Documentation

25.297.1.1 EStateID gdcm::network::ULActionDT2::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]

Implements gdcm::network::ULAction.

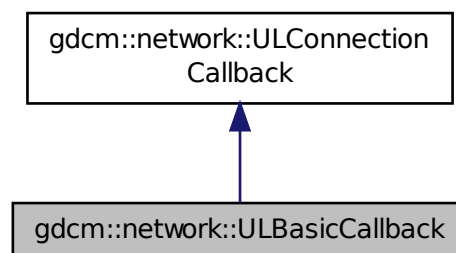
The documentation for this class was generated from the following file:

- gdcmULActionDT.h

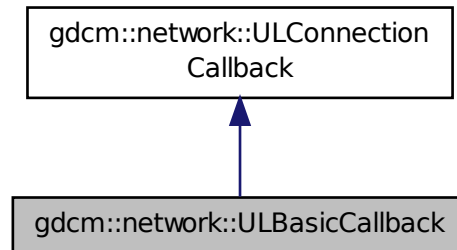
25.298 gdcm::network::ULBasicCallback Class Reference

```
#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`
- `virtual void HandleDataSet (const DataSet &inDataSet)`

Additional Inherited Members

25.298.1 Detailed Description

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`.

25.298.2 Constructor & Destructor Documentation

25.298.2.1 `gdcm::network::ULBasicCallback::ULBasicCallback ()` `[inline]`

25.298.2.2 `virtual gdcm::network::ULBasicCallback::~~ULBasicCallback ()` `[inline], [virtual]`

25.298.3 Member Function Documentation

25.298.3.1 `std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetDataSets () const`

25.298.3.2 `virtual void gdcm::network::ULBasicCallback::HandleDataSet (const DataSet & inDataSet)` `[virtual]`

Implements `gdcm::network::ULConnectionCallback`.

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

- `gdcmULBasicCallback.h`

25.299 gdcmm::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 <gdcmmULConnection.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 ()

25.299.1 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 `gdc`m 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.

25.299.2 Constructor & Destructor Documentation

25.299.2.1 `gdc`m::network::ULConnection::ULConnection (const `ULConnectionInfo` & *inUserInfo*)

25.299.2.2 virtual `gdc`m::network::ULConnection::~~ULConnection () [virtual]

25.299.3 Member Function Documentation

25.299.3.1 void `gdc`m::network::ULConnection::AddAcceptedPresentationContext (const `PresentationContextAC` & *inPC*)

25.299.3.2 `PresentationContextRQ` `gdc`m::network::ULConnection::FindContext (const `DataElement` & *de*) const

25.299.3.3 std::vector<`PresentationContextAC`> const& `gdc`m::network::ULConnection::GetAcceptedPresentationContexts () const

25.299.3.4 std::vector<`PresentationContextAC`>& `gdc`m::network::ULConnection::GetAcceptedPresentationContexts ()

25.299.3.5 const `ULConnectionInfo`& `gdc`m::network::ULConnection::GetConnectionInfo () const

25.299.3.6 uint32_t `gdc`m::network::ULConnection::GetMaxPDUSize () const

25.299.3.7 const `PresentationContextAC`* `gdc`m::network::ULConnection::GetPresentationContextACByID (uint8_t *id*) const

25.299.3.8 uint8_t `gdc`m::network::ULConnection::GetPresentationContextIDFromPresentationContext (`PresentationContextRQ` const & *pc*) const

return 0 upon error

25.299.3.9 const `PresentationContextRQ`* `gdc`m::network::ULConnection::GetPresentationContextRQByID (uint8_t *id*) const

25.299.3.10 std::vector<`PresentationContextRQ`> const& `gdc`m::network::ULConnection::GetPresentationContexts () const

25.299.3.11 std::iostream* `gdc`m::network::ULConnection::GetProtocol ()

25.299.3.12 `EStateID` `gdc`m::network::ULConnection::GetState () const

25.299.3.13 `ARTIMTimer`& `gdc`m::network::ULConnection::GetTimer ()

25.299.3.14 bool `gdc`m::network::ULConnection::InitializeConnection ()

used to establish scu connections

25.299.3.15 `bool gdcm::network::ULConnection::InitializeIncomingConnection ()`

used to establish scp connections

25.299.3.16 `void gdcm::network::ULConnection::SetMaxPDUSize (uint32_t inSize)`

25.299.3.17 `void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< PresentationContextRQ > & inContexts)`

25.299.3.18 `void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< PresentationContext > & inContexts)`

25.299.3.19 `void gdcm::network::ULConnection::SetState (const EStateID & inState)`

25.299.3.20 `void gdcm::network::ULConnection::StopProtocol ()`

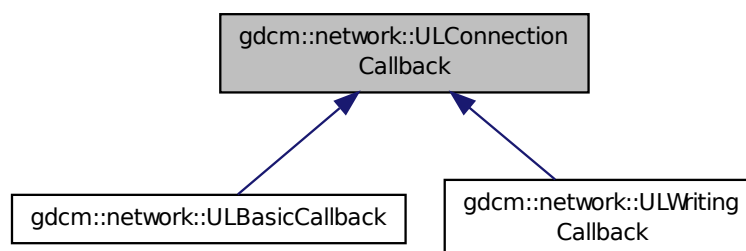
The documentation for this class was generated from the following file:

- gdcmULConnection.h

25.300 gdcm::network::ULConnectionCallback Class Reference

```
#include <gdcmULConnectionCallback.h>
```

Inheritance diagram for gdcm::network::ULConnectionCallback:



Public Member Functions

- `ULConnectionCallback ()`
- `~ULConnectionCallback ()`
- `bool DataSetHandles () const`
- `virtual void HandleDataSet (const DataSet &inDataSet)=0`
- `void ResetHandledDataSet ()`

Protected Member Functions

- void DataSetHandled ()

25.300.1 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 mHandledDataSet to true. otherwise, the cmove event loop handler will not know data was received, and proceed to end the loop prematurely.

25.300.2 Constructor & Destructor Documentation

25.300.2.1 `gdcm::network::ULConnectionCallback::ULConnectionCallback () [inline]`

25.300.2.2 `gdcm::network::ULConnectionCallback::~~ULConnectionCallback () [inline]`

25.300.3 Member Function Documentation

25.300.3.1 `void gdcm::network::ULConnectionCallback::DataSetHandled () [inline], [protected]`

25.300.3.2 `bool gdcm::network::ULConnectionCallback::DataSetHandles () const [inline]`

25.300.3.3 `virtual void gdcm::network::ULConnectionCallback::HandleDataSet (const DataSet & inDataSet) [pure virtual]`

Implemented in `gdcm::network::ULWritingCallback`, and `gdcm::network::ULBasicCallback`.

25.300.3.4 `void gdcm::network::ULConnectionCallback::ResetHandledDataSet () [inline]`

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

- `gdcmULConnectionCallback.h`

25.301 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
- UserInformation GetUserInformation () const
- bool Initialize (UserInformation inUserInformation, const char inCalledAETitle[16], const char inCallingAETitle[16], unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void SetMaxPDULength (unsigned long inMaxPDULength)

25.301.1 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.

25.301.2 Constructor & Destructor Documentation

25.301.2.1 gdcmm::network::ULConnectionInfo::ULConnectionInfo ()

25.301.3 Member Function Documentation

25.301.3.1 const char* gdcmm::network::ULConnectionInfo::GetCalledAETitle () const

25.301.3.2 std::string gdcmm::network::ULConnectionInfo::GetCalledComputerName () const

25.301.3.3 unsigned long gdcmm::network::ULConnectionInfo::GetCalledIPAddress () const

25.301.3.4 int gdcmm::network::ULConnectionInfo::GetCalledIPPort () const

25.301.3.5 const char* gdcmm::network::ULConnectionInfo::GetCallingAETitle () const

25.301.3.6 unsigned long gdcmm::network::ULConnectionInfo::GetMaxPDULength () const

25.301.3.7 UserInformation gdcmm::network::ULConnectionInfo::GetUserInformation () const

25.301.3.8 bool gdcmm::network::ULConnectionInfo::Initialize (UserInformation inUserInformation, const char inCalledAETitle[16], const char inCallingAETitle[16], unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)

25.301.3.9 void gdcmm::network::ULConnectionInfo::SetMaxPDULength (unsigned long inMaxPDULength)

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

- gdcmmULConnectionInfo.h

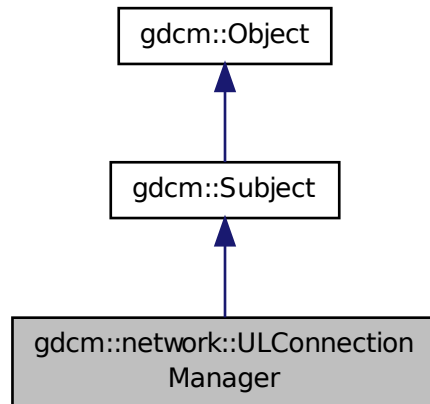
25.302 gdcmm::network::ULConnectionManager Class Reference

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

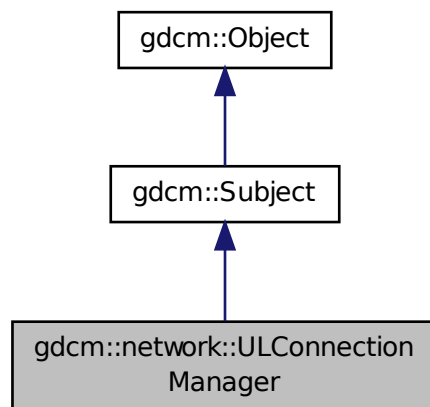
across the connection, etc).

```
#include <gdcmULConnectionManager.h>
```

Inheritance diagram for `gdcm::network::ULConnectionManager`:



Collaboration diagram for `gdcm::network::ULConnectionManager`:



Public Member Functions

- `ULConnectionManager()`

- ~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)
- void SendMove (const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendStore (const File &file)
- void SendStore (const File &file, ULConnectionCallback *inCallback)

callback based API

25.302.1 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.

25.302.2 Constructor & Destructor Documentation

25.302.2.1 gdcmm::network::ULConnectionManager::ULConnectionManager ()

25.302.2.2 gdcmm::network::ULConnectionManager::~~ULConnectionManager ()

25.302.3 Member Function Documentation

25.302.3.1 bool gdcmm::network::ULConnectionManager::BreakConnection (const double & inTimeout)

25.302.3.2 void gdcmm::network::ULConnectionManager::BreakConnectionNow ()

25.302.3.3 bool gdcmm::network::ULConnectionManager::EstablishConnection (const std::string & inAETitle, const std::string & inConnectAETitle, const std::string & inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, std::vector< PresentationContext > const & pcVector)

returns true if a connection of the given AETitle (ie, 'this' program) is able to connect to the given AETitle and Port in a certain amount of time providing the connection type will establish the proper exchange syntax with a server; if a different functionality is required, a different connection should be established. returns false if the connection type is 'move'— have to give a return port for move to work as specified.

25.302.3.4 `bool gdcmm::network::ULConnectionManager::EstablishConnectionMove (const std::string & inAETitle, const std::string & inConnectAETitle, const std::string & inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, uint16_t inReturnPort, std::vector< PresentationContext > const & pcVector)`

returns true for above reasons, but contains the special 'move' port

25.302.3.5 `std::vector<PresentationDataValue> gdcmm::network::ULConnectionManager::SendEcho ()`

25.302.3.6 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendFind (const BaseRootQuery * inRootQuery)`

25.302.3.7 `void gdcmm::network::ULConnectionManager::SendFind (const BaseRootQuery * inRootQuery, ULConnectionCallback * inCallback)`

25.302.3.8 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendMove (const BaseRootQuery * inRootQuery)`

25.302.3.9 `void gdcmm::network::ULConnectionManager::SendMove (const BaseRootQuery * inRootQuery, ULConnectionCallback * inCallback)`

25.302.3.10 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendStore (const File & file)`

25.302.3.11 `void gdcmm::network::ULConnectionManager::SendStore (const File & file, ULConnectionCallback * inCallback)`

callback based API

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

- `gdcmmULConnectionManager.h`

25.303 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)`
- `ULEvent (const EEventID &inEventID, BasePDU *inBasePDU)`
- `~ULEvent ()`
- `EEventID GetEvent () const`
- `std::vector< BasePDU * > GetPDUs () const`
- `void SetEvent (const EEventID &inEvent)`
- `void SetPDU (std::vector< BasePDU * > inPDU)`

25.303.1 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!

25.303.2 Constructor & Destructor Documentation

25.303.2.1 `gdcmm::network::ULEvent::ULEvent (const EEventID & inEventID, std::vector< BasePDU * > inBasePDU) [inline]`

25.303.2.2 `gdcmm::network::ULEvent::ULEvent (const EEventID & inEventID, BasePDU * inBasePDU) [inline]`

25.303.2.3 `gdcmm::network::ULEvent::~~ULEvent () [inline]`

25.303.3 Member Function Documentation

25.303.3.1 `EEventID gdcmm::network::ULEvent::GetEvent () const [inline]`

25.303.3.2 `std::vector<BasePDU*> gdcmm::network::ULEvent::GetPDUs () const [inline]`

25.303.3.3 `void gdcmm::network::ULEvent::SetEvent (const EEventID & inEvent) [inline]`

25.303.3.4 `void gdcmm::network::ULEvent::SetPDU (std::vector< BasePDU * > inPDU) [inline]`

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

- gdcmmULEvent.h

25.304 gdcmm::network::ULTransitionTable Class Reference

ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

```
#include <gdcmmULTransitionTable.h>
```

Public Member Functions

- ULTransitionTable ()
- void HandleEvent (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, E-EventID &outRaisedEvent) const
- void PrintTable () const

25.304.1 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 Transition-Table 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.

25.304.2 Constructor & Destructor Documentation

25.304.2.1 `gdcmm::network::ULTransitionTable::ULTransitionTable ()`

25.304.3 Member Function Documentation

25.304.3.1 `void gdcmm::network::ULTransitionTable::HandleEvent (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) const`

25.304.3.2 `void gdcmm::network::ULTransitionTable::PrintTable () const`

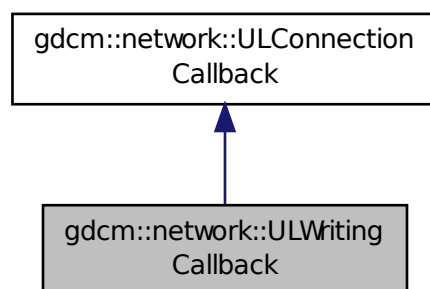
The documentation for this class was generated from the following file:

- `gdcmmULTransitionTable.h`

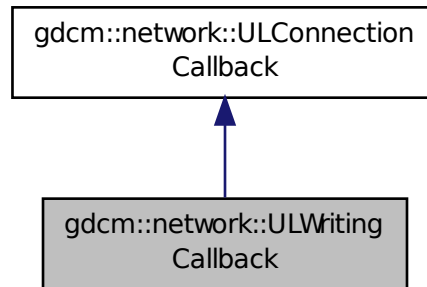
25.305 `gdcmm::network::ULWritingCallback` Class Reference

```
#include <gdcmmULWritingCallback.h>
```

Inheritance diagram for `gdcmm::network::ULWritingCallback`:



Collaboration diagram for gdcm::network::ULWritingCallback:



Public Member Functions

- `ULWritingCallback ()`
- `virtual ~ULWritingCallback ()`
- `virtual void HandleDataSet (const DataSet &inDataSet)`
- `void SetDirectory (const std::string &inDirectoryName)`

provide the directory into which all files are written.

Additional Inherited Members

25.305.1 Constructor & Destructor Documentation

25.305.1.1 `gdcm::network::ULWritingCallback::ULWritingCallback ()` `[inline]`

25.305.1.2 `virtual gdcm::network::ULWritingCallback::~~ULWritingCallback ()` `[inline]`, `[virtual]`

25.305.2 Member Function Documentation

25.305.2.1 `virtual void gdcm::network::ULWritingCallback::HandleDataSet (const DataSet & inDataSet)` `[virtual]`

Implements `gdcm::network::ULConnectionCallback`.

25.305.2.2 `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:

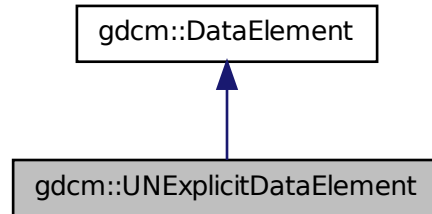
- `gdcmULWritingCallback.h`

25.306 gdcm::UNExplicitDataElement Class Reference

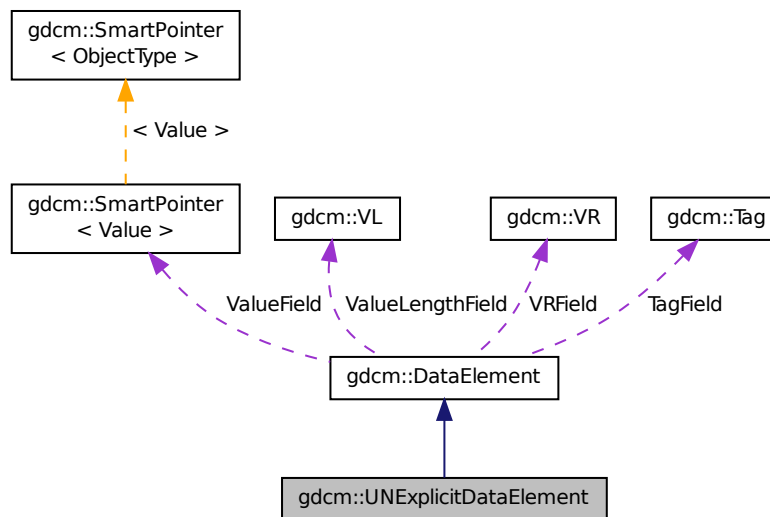
Class to read/write a DataElement as UNExplicit Data Element.

```
#include <gdcmUNExplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitDataElement:



Collaboration diagram for gdcm::UNExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap >
std::istream & Read (std::istream &is)

- `template<typename TSwap >`
`std::istream & ReadPreValue (std::istream &is)`
- `template<typename TSwap >`
`std::istream & ReadValue (std::istream &is)`
- `template<typename TSwap >`
`std::istream & ReadWithLength (std::istream &is, VL &length)`

Additional Inherited Members

25.306.1 Detailed Description

Class to read/write a DataElement as UNExplicit Data Element.

Note

bla

25.306.2 Member Function Documentation

25.306.2.1 VL gdcm::UNExplicitDataElement::GetLength () const

Reimplemented from `gdcm::DataElement`.

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

Reimplemented from `gdcm::DataElement`.

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

25.306.2.4 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadValue (std::istream & is)`

25.306.2.5 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

Reimplemented from `gdcm::DataElement`.

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

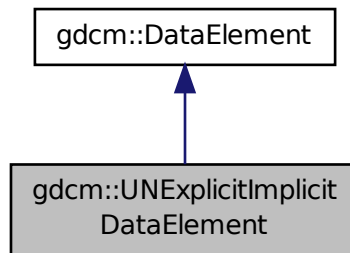
- `gdcmUNExplicitDataElement.h`

25.307 gdcm::UNExplicitImplicitDataElement Class Reference

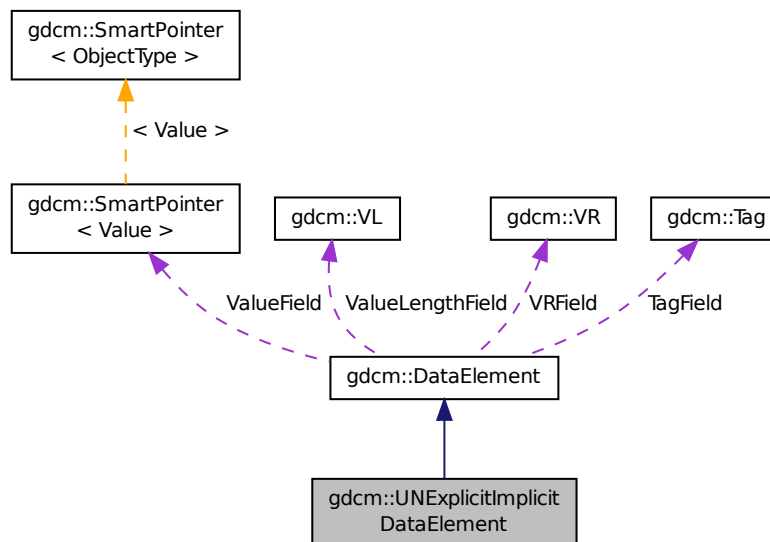
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:

```
#include <gdcmUNExplicitImplicitDataElement.h>
```

Inheritance diagram for `gdc::UNExplicitImplicitDataElement`:



Collaboration diagram for `gdc::UNExplicitImplicitDataElement`:



Public Member Functions

- `VL GetLength () const`
- `template<typename TSwap > std::istream & Read (std::istream &is)`
- `template<typename TSwap > std::istream & ReadPreValue (std::istream &is)`
- `template<typename TSwap > std::istream & ReadValue (std::istream &is)`

Additional Inherited Members

25.307.1 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

25.307.2 Member Function Documentation

25.307.2.1 VL gdcm::UNExplicitImplicitDataElement::GetLength () const

Reimplemented from gdcm::DataElement.

25.307.2.2 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::Read (std::istream & is)

Reimplemented from gdcm::DataElement.

25.307.2.3 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue (std::istream & is)

25.307.2.4 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadValue (std::istream & is)

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

- gdcmUNExplicitImplicitDataElement.h

25.308 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)

25.308.1 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

25.308.2 Member Function Documentation

25.308.2.1 `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 $(n / 2) * 3$

25.308.2.2 `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

25.309 gdcm::Usage Class Reference

Usage.

```
#include <gdcmUsage.h>
```

Public Types

- enum UsageType {
Mandatory,
Conditional,
UserOption,
Invalid }

Public Member Functions

- Usage (UsageType type=Invalid)
- operator UsageType () const

Static Public Member Functions

- static const char * GetUsageString (UsageType type)
- static UsageType GetUsageType (const char *type)

Friends

- `std::ostream & operator<< (std::ostream &os, const Usage &vr)`

25.309.1 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.

25.309.2 Member Enumeration Documentation

25.309.2.1 `enum gdcm::Usage::UsageType`

Enumerator:

Mandatory

Conditional

UserOption

Invalid

25.309.3 Constructor & Destructor Documentation

25.309.3.1 `gdcm::Usage::Usage (UsageType type = Invalid) [inline]`

25.309.4 Member Function Documentation

25.309.4.1 `static const char* gdcm::Usage::GetUsageString (UsageType type) [static]`

Referenced by `gdcm::operator<<()`.

25.309.4.2 `static UsageType gdcM::Usage::GetUsageType (const char * type)` `[static]`

25.309.4.3 `gdcM::Usage::operator UsageType () const` `[inline]`

25.309.5 Friends And Related Function Documentation

25.309.5.1 `std::ostream& operator<< (std::ostream & os, const Usage & vr)` `[friend]`

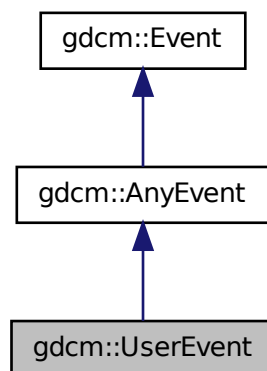
The documentation for this class was generated from the following file:

- `gdcMUsage.h`

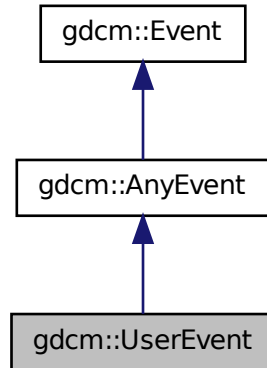
25.310 gdcM::UserEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for `gdcM::UserEvent`:



Collaboration diagram for gdcm::UserEvent:



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

- gdcmEvent.h

25.311 gdcm::network::UserInfo Class Reference

UserInfo Table 9-16 USER INFORMATION ITEM FIELDS.

```
#include <gdcmUserInfo.h>
```

Public Member Functions

- UserInfo ()
- const MaximumLengthSub & GetMaximumLengthSub () const
- MaximumLengthSub & GetMaximumLengthSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

25.311.1 Detailed Description

UserInfo Table 9-16 USER INFORMATION ITEM FIELDS.

TODO what is the goal of :

Table 9-20 USER INFORMATION ITEM FIELDS

25.311.2 Constructor & Destructor Documentation

25.311.2.1 `gdcm::network::UserInformation::UserInformation ()`

25.311.3 Member Function Documentation

25.311.3.1 `const MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub () const` `[inline]`

25.311.3.2 `MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ()` `[inline]`

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

25.311.3.4 `std::istream& gdcm::network::UserInformation::Read (std::istream & is)`

25.311.3.5 `size_t gdcm::network::UserInformation::Size () const`

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

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

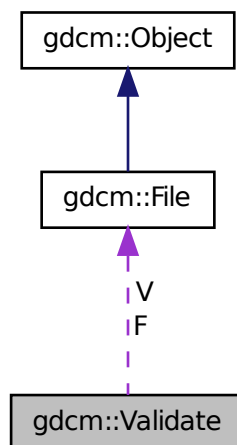
- `gdcmUserInformation.h`

25.312 gdcm::Validate Class Reference

Validate class.

```
#include <gdcmValidate.h>
```

Collaboration diagram for `gdcm::Validate`:



Public Member Functions

- `Validate ()`
- `~Validate ()`
- `const File & GetValidatedFile ()`
- `void SetFile (File const &f)`
- `void Validation ()`

Protected Attributes

- `const File * F`
- `File V`

25.312.1 Detailed Description

Validate class.

25.312.2 Constructor & Destructor Documentation

25.312.2.1 `gdcm::Validate::Validate ()`

25.312.2.2 `gdcm::Validate::~~Validate ()`

25.312.3 Member Function Documentation

25.312.3.1 `const File& gdcm::Validate::GetValidatedFile ()` `[inline]`

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

25.312.3.3 `void gdcm::Validate::Validation ()`

25.312.4 Member Data Documentation

25.312.4.1 `const File* gdcm::Validate::F` `[protected]`

25.312.4.2 `File gdcm::Validate::V` `[protected]`

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

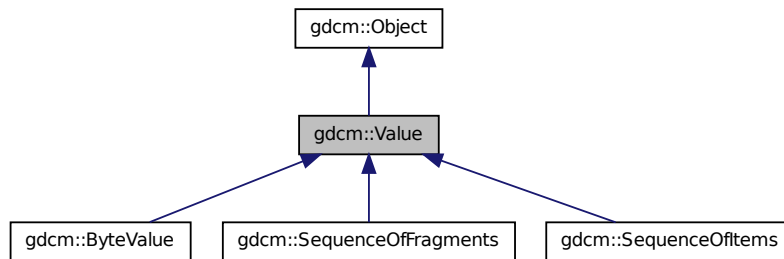
- `gdcmValidate.h`

25.313 gdcm::Value Class Reference

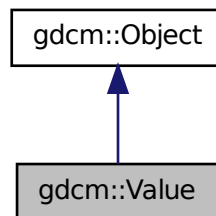
Class to represent the value of a Data Element.

```
#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`

Additional Inherited Members

25.313.1 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.

25.313.2 Constructor & Destructor Documentation

25.313.2.1 `gdcm::Value::Value () [inline]`

25.313.2.2 `gdcm::Value::~~Value () [inline]`

25.313.3 Member Function Documentation

25.313.3.1 `virtual void gdcm::Value::Clear () [pure virtual]`

Implemented in `gdcm::ByteValue`, `gdcm::SequenceOfItems`, and `gdcm::SequenceOfFragments`.

25.313.3.2 `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()`.

25.313.3.3 `virtual bool gdcm::Value::operator== (const Value & val) const [pure virtual]`

Implemented in `gdcm::SequenceOfItems`, `gdcm::SequenceOfFragments`, and `gdcm::ByteValue`.

25.313.3.4 `virtual void gdcm::Value::SetLength (VL /) [pure virtual]`

Implemented in `gdcm::ByteValue`, `gdcm::SequenceOfItems`, and `gdcm::SequenceOfFragments`.

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

- `gdcmValue.h`

25.314 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)`
- static `const std::ostream & Write (std::ostream &os, const Value &v)`

25.314.1 Detailed Description

```
template<typename TDE, typename TSwap, typename TType = uint8_t>class gdcm::ValueIO< TDE, TSwap, TType >
```

Class to dispatch template calls.

25.314.2 Member Function Documentation

25.314.2.1 `template<typename TDE , typename TSwap , typename TType = uint8_t> static std::istream& gdcmm::ValueIO< TDE, TSwap, TType >::Read (std::istream & is, Value & v) [static]`

25.314.2.2 `template<typename TDE , typename TSwap , typename TType = uint8_t> static const std::ostream& gdcmm::ValueIO< TDE, TSwap, TType >::Write (std::ostream & os, const Value & v) [static]`

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

- `gdcmmValueIO.h`

25.315 gdcmm::Version Class Reference

major/minor and build version

```
#include <gdcmmVersion.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)`

25.315.1 Detailed Description

major/minor and build version

25.315.2 Constructor & Destructor Documentation

25.315.2.1 `gdcmm::Version::Version () [inline]`

25.315.2.2 `gdcmm::Version::~~Version () [inline]`

25.315.3 Member Function Documentation

- 25.315.3.1 `static int gdcm::Version::GetBuildVersion () [static]`
- 25.315.3.2 `static int gdcm::Version::GetMajorVersion () [static]`
- 25.315.3.3 `static int gdcm::Version::GetMinorVersion () [static]`
- 25.315.3.4 `static const char* gdcm::Version::GetVersion () [static]`
- 25.315.3.5 `void gdcm::Version::Print (std::ostream & os = std::cout) const`

Referenced by `gdcm::operator<<()`.

25.315.4 Friends And Related Function Documentation

- 25.315.4.1 `std::ostream& operator<< (std::ostream & os, const Version & v) [friend]`

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

- `gdcmVersion.h`

25.316 gdcm::VL Class Reference

Value Length.

```
#include <gdcmVL.h>
```

Public Types

- `typedef uint32_t Type`

Public Member Functions

- `VL (uint32_t vl=0)`
- `VL GetLength () const`
- `bool IsOdd () const`
Return whether or not the VL is odd or not.
- `bool IsUndefined () const`
- `operator uint32_t () const`
- `VL & operator++ ()`
- `VL operator++ (int)`
- `VL & operator+= (VL const &vl)`
+= operator
- `template<typename TSwap >`
`std::istream & Read (std::istream &is)`
- `template<typename TSwap >`
`std::istream & Read16 (std::istream &is)`
- `void SetToUndefined ()`
- `template<typename TSwap >`
`const std::ostream & Write (std::ostream &os) const`

- `template<typename TSwap >`
`const std::ostream & Write16 (std::ostream &os) const`

Static Public Member Functions

- `static uint16_t GetVL16Max ()`
- `static uint32_t GetVL32Max ()`

Friends

- `std::ostream & operator<< (std::ostream &os, const VL &vl)`

25.316.1 Detailed Description

Value Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

25.316.2 Member Typedef Documentation

25.316.2.1 `typedef uint32_t gdcm::VL::Type`

25.316.3 Constructor & Destructor Documentation

25.316.3.1 `gdcm::VL::VL (uint32_t vl = 0) [inline]`

25.316.4 Member Function Documentation

25.316.4.1 `VL gdcm::VL::GetLength () const [inline]`

Referenced by `gdcm::FileMetaInformation::GetFullLength()`, `gdcm::Fragment::GetLength()`, and `gdcm::Item::Write()`.

25.316.4.2 `static uint16_t gdcm::VL::GetVL16Max () [inline], [static]`

25.316.4.3 `static uint32_t gdcm::VL::GetVL32Max () [inline], [static]`

25.316.4.4 `bool gdcm::VL::IsOdd () const [inline]`

Return whether or not the VL is odd or not.

Referenced by `gdcm::ByteValue::SetLength()`.

25.316.4.5 `bool gdcm::VL::IsUndefined () const [inline]`

Referenced by `gdcm::ByteValue::SetLength()`.

25.316.4.6 `gdcm::VL::operator uint32_t () const` `[inline]`

25.316.4.7 `VL& gdcm::VL::operator++ ()` `[inline]`

25.316.4.8 `VL gdcm::VL::operator++ (int)` `[inline]`

25.316.4.9 `VL& gdcm::VL::operator+=(VL const & vl)` `[inline]`

`+=` operator

25.316.4.10 `template<typename TSwap> std::istream& gdcm::VL::Read (std::istream & is)` `[inline]`

Referenced by `gdcm::Fragment::Read()`.

25.316.4.11 `template<typename TSwap> std::istream& gdcm::VL::Read16 (std::istream & is)` `[inline]`

25.316.4.12 `void gdcm::VL::SetToUndefined ()` `[inline]`

25.316.4.13 `template<typename TSwap> const std::ostream& gdcm::VL::Write (std::ostream & os) const` `[inline]`

Referenced by `gdcm::Fragment::Write()`, `gdcm::SequenceOfFragments::Write()`, `gdcm::SequenceOfItems::Write()`, and `gdcm::Item::Write()`.

25.316.4.14 `template<typename TSwap> const std::ostream& gdcm::VL::Write16 (std::ostream & os) const` `[inline]`

25.316.5 Friends And Related Function Documentation

25.316.5.1 `std::ostream& operator<< (std::ostream & os, const VL & vl)` `[friend]`

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

- `gdcmVL.h`

25.317 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)

25.317.1 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

25.317.2 Member Enumeration Documentation

25.317.2.1 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

25.317.3 Constructor & Destructor Documentation

25.317.3.1 `gdcm::VM::VM (VMType type = VM0) [inline]`

25.317.4 Member Function Documentation

25.317.4.1 `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.

25.317.4.2 `static unsigned int gdcm::VM::GetIndex (VMType vm) [static], [protected]`

25.317.4.3 `unsigned int gdcm::VM::GetLength () const`

25.317.4.4 `static unsigned int gdcm::VM::GetNumberOfElementsFromArray (const char * array, unsigned int length) [static]`

25.317.4.5 `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<<()`.

25.317.4.6 static VMType gdcm::VM::GetVMType (const char * *vm*) [static]

25.317.4.7 static VMType gdcm::VM::GetVMTypeFromLength (unsigned int *length*, unsigned int *size*) [static]

25.317.4.8 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

25.317.4.9 gdcm::VM::operator VMType () const [inline]

25.317.5 Friends And Related Function Documentation

25.317.5.1 std::ostream& operator<< (std::ostream & *os*, const VM & *vm*) [friend]

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

- gdcmVM.h

25.318 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,
 - 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 | 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 | OF | OW | SL | SQ | SS | UL | UN | US,
 - VR_VM1 = AS | LT | ST | UT | SQ | OF | 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)`

25.318.1 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.

25.318.2 Member Enumeration Documentation

25.318.2.1 enum gdcm::VR::VRType

Enumerator:

INVALID
AE
AS
AT
CS
DA

DS
DT
FD
FL
IS
LO
LT
OB
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

25.318.3 Constructor & Destructor Documentation

25.318.3.1 `gdcmm::VR::VR (VRType vr = INVALID) [inline]`

25.318.4 Member Function Documentation

25.318.4.1 `static bool gdcmm::VR::CanDisplay (VRType vr) [static]`

25.318.4.2 `bool gdcmm::VR::Compatible (VR const & vr) const`

25.318.4.3 `int gdcm::VR::GetLength () const [inline]`

25.318.4.4 `static uint32_t gdcm::VR::GetLength (VRType vr) [inline], [static]`

25.318.4.5 `unsigned int gdcm::VR::GetSize () const [inline]`

References AE, US_SS, and VRTypeTemplateCase.

25.318.4.6 `unsigned int gdcm::VR::GetSizeof () const`

25.318.4.7 `static const char* gdcm::VR::GetVRString (VRType vr) [static]`

Referenced by `gdcm::operator<<()`.

25.318.4.8 `static const char* gdcm::VR::GetVRStringFromFile (VRType vr) [static]`

25.318.4.9 `static VRType gdcm::VR::GetVRType (const char * vr) [static]`

25.318.4.10 `static VRType gdcm::VR::GetVRTypeFromFile (const char * vr) [static]`

25.318.4.11 `static bool gdcm::VR::IsASCII (VRType vr) [static]`

25.318.4.12 `static bool gdcm::VR::IsASCII2 (VRType vr) [static]`

25.318.4.13 `static bool gdcm::VR::IsBinary (VRType vr) [static]`

25.318.4.14 `static bool gdcm::VR::IsBinary2 (VRType vr) [static]`

25.318.4.15 `bool gdcm::VR::IsDual () const`

25.318.4.16 `static bool gdcm::VR::IsSwap (const char * vr) [static]`

25.318.4.17 `static bool gdcm::VR::IsValid (const char * vr) [static]`

25.318.4.18 `static bool gdcm::VR::IsValid (const char * vr1, VRType vr2) [static]`

25.318.4.19 `bool gdcm::VR::IsVRFile () const`

Referenced by `gdcm::DataElement::SetVR()`.

25.318.4.20 `gdcm::VR::operator VRType () const [inline]`

25.318.4.21 `std::istream& gdcm::VR::Read (std::istream & is) [inline]`

References `gdcmDebugMacro`, `INVALID`, and `VR_END`.

25.318.4.22 `const std::ostream& gdcm::VR::Write (std::ostream & os) const [inline]`

References `gdcmAssertAlwaysMacro`, and `INVALID`.

25.318.5 Friends And Related Function Documentation

25.318.5.1 `std::ostream& operator<< (std::ostream & os, const VR & vr)` [friend]

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

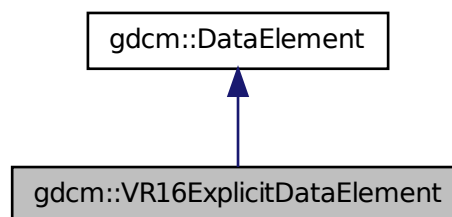
- `gdcmVR.h`

25.319 `gdcm::VR16ExplicitDataElement` Class Reference

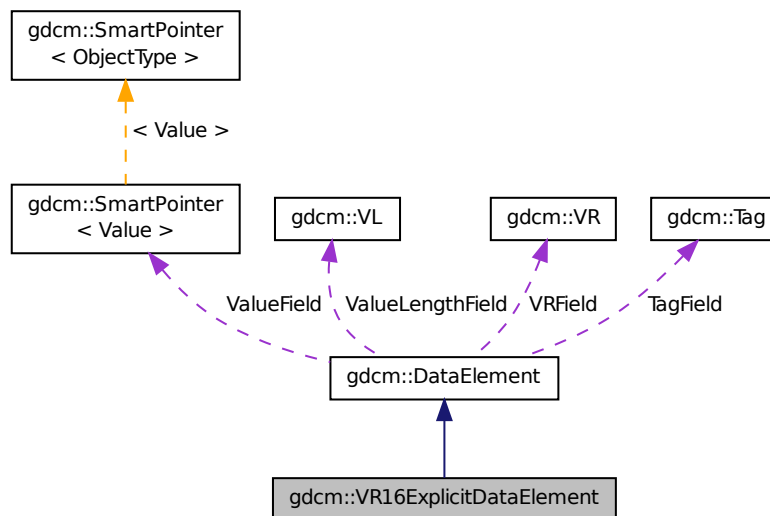
Class to read/write a DataElement as Explicit Data Element.

```
#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)
- template<typename TSwap >
std::istream & ReadWithLength (std::istream &is, VL &length)

Additional Inherited Members

25.319.1 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

25.319.2 Member Function Documentation

25.319.2.1 VL gdcm::VR16ExplicitDataElement::GetLength () const

Reimplemented from gdcm::DataElement.

25.319.2.2 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::Read (std::istream & is)`

Reimplemented from `gdcm::DataElement`.

25.319.2.3 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (std::istream & is)`

25.319.2.4 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadValue (std::istream & is)`

25.319.2.5 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

Reimplemented from `gdcm::DataElement`.

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

- `gdcmVR16ExplicitDataElement.h`

25.320 `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)`

25.320.1 Member Function Documentation

25.320.1.1 `static uint16_t gdcm::VRVLSIZE< 0 >::Read (std::istream & is)` `[inline]`, `[static]`

25.320.1.2 `static void gdcm::VRVLSIZE< 0 >::Write (std::ostream & os)` `[inline]`, `[static]`

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

- `gdcmAttribute.h`

25.321 `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)`

25.321.1 Member Function Documentation

25.321.1.1 `static uint32_t gdcmm::VRVLSize< 1 >::Read (std::istream & is) [inline], [static]`

25.321.1.2 `static void gdcmm::VRVLSize< 1 >::Write (std::ostream & os) [inline], [static]`

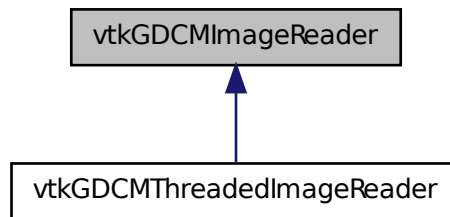
The documentation for this class was generated from the following file:

- gdcmmAttribute.h

25.322 vtkGDCMImageReader Class Reference

```
#include <vtkGDCMImageReader.h>
```

Inheritance 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
- 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 gdcmm::ImageReader &reader)
- int LoadSingleFile (const char *filename, char *pointer, unsigned long &outlen)
- int RequestDataCompat ()
- int RequestInformationCompat ()
- void SetFilePattern (const char *)
- void SetFilePrefix (const char *)
- vtkGetStringMacro (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

25.322.1 Detailed Description

Examples:

AWTMedical3.java, Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmmorthoplanes.cxx, gdcmmreslice.cxx, gdcmmtexture.cxx, gdcmmvolume.cxx, HelloActiviz.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MagnifyFile.cxx, MetalImageMD5Activiz.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, offscreenimage.cxx, ReadSeriesIntoVTK.java, RefCounting.cs, and reslicesphere.cxx.

25.322.2 Constructor & Destructor Documentation

25.322.2.1 `vtkGDCMImageReader::vtkGDCMImageReader ()` [protected]

25.322.2.2 `vtkGDCMImageReader::~~vtkGDCMImageReader ()` [protected]

25.322.3 Member Function Documentation

25.322.3.1 `virtual int vtkGDCMImageReader::CanReadFile (const char * fname)` [virtual]

Examples:

MetalImageMD5Activiz.cs.

25.322.3.2 `void vtkGDCMImageReader::ExecuteData (vtkDataObject * out)` [protected]

Reimplemented in `vtkGDCMThreadedImageReader`.

25.322.3.3 `void vtkGDCMImageReader::ExecuteInformation ()` [protected]

Reimplemented in `vtkGDCMThreadedImageReader`.

- 25.322.3.4 `void vtkGDCMImageReader::FillMedicalImageInformation (const gdcm::ImageReader & reader)`
[protected]
- 25.322.3.5 `virtual const char* vtkGDCMImageReader::GetDescriptiveName ()` [inline],[virtual]
- 25.322.3.6 `virtual const char* vtkGDCMImageReader::GetFileExtensions ()` [inline],[virtual]
- 25.322.3.7 `vtkImageData* vtkGDCMImageReader::GetIconImage ()`
- 25.322.3.8 `vtkImageData* vtkGDCMImageReader::GetOverlay (int i)`
- 25.322.3.9 `int vtkGDCMImageReader::LoadSingleFile (const char * filename, char * pointer, unsigned long & outlen)`
[protected]
- 25.322.3.10 `static vtkGDCMImageReader* vtkGDCMImageReader::New ()` [static]

Reimplemented in `vtkGDCMThreadedImageReader`.

Examples:

`Convert16BitsTo8Bits.cxx`, `ConvertMultiFrameToSingleFrame.cxx`, `ConvertRGBToLuminance.cxx`, `ConvertSingleBitTo8Bits.cxx`, `gdcmorthoplanes.cxx`, `gdcmreslice.cxx`, `gdcmtexture.cxx`, `gdcmvolume.cxx`, `HelloActiviz.cs`, `HelloActiviz3.cs`, `HelloActiviz4.cs`, `HelloActiviz5.cs`, `HelloVTKWorld.cs`, `MagnifyFile.cxx`, `MetalImageMD5Activiz.cs`, `offscreenimage.cxx`, `RefCounting.cs`, and `reslicesphere.cxx`.

- 25.322.3.11 `virtual void vtkGDCMImageReader::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

Reimplemented in `vtkGDCMThreadedImageReader`.

- 25.322.3.12 `int vtkGDCMImageReader::RequestDataCompat ()` [protected]

Reimplemented in `vtkGDCMThreadedImageReader`.

- 25.322.3.13 `int vtkGDCMImageReader::RequestInformationCompat ()` [protected]

- 25.322.3.14 `virtual void vtkGDCMImageReader::SetCurve (vtkPolyData * pd)` [virtual]

- 25.322.3.15 `virtual void vtkGDCMImageReader::SetFileNames (vtkStringArray *)` [virtual]

Examples:

`gdcmorthoplanes.cxx`, `HelloActiviz3.cs`, `HelloActiviz4.cs`, `HelloActiviz5.cs`, `MIPViewer.java`, `MPRViewer.java`, `MPRViewer2.java`, and `ReadSeriesIntoVTK.java`.

- 25.322.3.16 `void vtkGDCMImageReader::SetFilePattern (const char *)` [inline],[protected]

- 25.322.3.17 `void vtkGDCMImageReader::SetFilePrefix (const char *)` [inline],[protected]

- 25.322.3.18 `virtual void vtkGDCMImageReader::SetMedicalImageProperties (vtkMedicalImageProperties * pd)` [virtual]

- 25.322.3.19 `vtkGDCMImageReader::vtkBooleanMacro (LoadOverlays , int)`
- 25.322.3.20 `vtkGDCMImageReader::vtkBooleanMacro (LoadIconImage , int)`
- 25.322.3.21 `vtkGDCMImageReader::vtkBooleanMacro (LossyFlag , int)`
- 25.322.3.22 `vtkGDCMImageReader::vtkBooleanMacro (ApplyLookupTable , int)`
- 25.322.3.23 `int vtkGDCMImageReader::vtkBooleanMacro (ApplyYBRToRGB , int)`
- 25.322.3.24 `vtkGDCMImageReader::vtkGetMacro (LoadOverlays , int)`
- 25.322.3.25 `vtkGDCMImageReader::vtkGetMacro (LoadIconImage , int)`
- 25.322.3.26 `vtkGDCMImageReader::vtkGetMacro (LossyFlag , int)`
- 25.322.3.27 `vtkGDCMImageReader::vtkGetMacro (NumberOfOverlays , int)`
- 25.322.3.28 `vtkGDCMImageReader::vtkGetMacro (NumberOfIconImages , int)`
- 25.322.3.29 `vtkGDCMImageReader::vtkGetMacro (ApplyLookupTable , int)`
- 25.322.3.30 `vtkGDCMImageReader::vtkGetMacro (ApplyYBRToRGB , int)`
- 25.322.3.31 `vtkGDCMImageReader::vtkGetMacro (ImageFormat , int)`
- 25.322.3.32 `vtkGDCMImageReader::vtkGetMacro (PlanarConfiguration , int)`
- 25.322.3.33 `vtkGDCMImageReader::vtkGetMacro (Shift , double)`
- 25.322.3.34 `vtkGDCMImageReader::vtkGetMacro (Scale , double)`
- 25.322.3.35 `vtkGDCMImageReader::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)`
- 25.322.3.36 `vtkGDCMImageReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)`
- 25.322.3.37 `vtkGDCMImageReader::vtkGetObjectMacro (FileNames , vtkStringArray)`
- 25.322.3.38 `vtkGDCMImageReader::vtkGetObjectMacro (Curve , vtkPolyData)`
- 25.322.3.39 `vtkGDCMImageReader::vtkGetStringMacro (FilePrefix) [protected]`
- 25.322.3.40 `vtkGDCMImageReader::vtkGetStringMacro (FilePattern) [protected]`
- 25.322.3.41 `vtkGDCMImageReader::vtkGetVector3Macro (ImagePositionPatient , double)`
- 25.322.3.42 `vtkGDCMImageReader::vtkGetVector6Macro (ImageOrientationPatient , double)`
- 25.322.3.43 `vtkGDCMImageReader::vtkSetMacro (LoadOverlays , int)`
- 25.322.3.44 `vtkGDCMImageReader::vtkSetMacro (LoadIconImage , int)`

- 25.322.3.45 `vtkGDCMImageReader::vtkSetMacro (LossyFlag , int)`
- 25.322.3.46 `vtkGDCMImageReader::vtkSetMacro (ApplyLookupTable , int)`
- 25.322.3.47 `vtkGDCMImageReader::vtkSetVector6Macro (ImageOrientationPatient , double)` [protected]
- 25.322.3.48 `vtkGDCMImageReader::vtkTypeRevisionMacro (vtkGDCMImageReader , vtkMedicalImageReader2)`

25.322.4 Member Data Documentation

- 25.322.4.1 `int vtkGDCMImageReader::ApplyInverseVideo` [protected]
- 25.322.4.2 `int vtkGDCMImageReader::ApplyLookupTable` [protected]
- 25.322.4.3 `int vtkGDCMImageReader::ApplyPlanarConfiguration` [protected]
- 25.322.4.4 `int vtkGDCMImageReader::ApplyShiftScale` [protected]
- 25.322.4.5 `int vtkGDCMImageReader::ApplyYBRToRGB` [protected]
- 25.322.4.6 `vtkPolyData* vtkGDCMImageReader::Curve` [protected]
- 25.322.4.7 `vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines` [protected]
- 25.322.4.8 `vtkStringArray* vtkGDCMImageReader::FileNames` [protected]
- 25.322.4.9 `int vtkGDCMImageReader::ForceRescale` [protected]
- 25.322.4.10 `int vtkGDCMImageReader::IconDataScalarType` [protected]
- 25.322.4.11 `int vtkGDCMImageReader::IconImageDataExtent[6]` [protected]
- 25.322.4.12 `int vtkGDCMImageReader::IconNumberOfScalarComponents` [protected]
- 25.322.4.13 `int vtkGDCMImageReader::ImageFormat` [protected]
- 25.322.4.14 `double vtkGDCMImageReader::ImageOrientationPatient[6]` [protected]
- 25.322.4.15 `double vtkGDCMImageReader::ImagePositionPatient[3]` [protected]
- 25.322.4.16 `int vtkGDCMImageReader::LoadIconImage` [protected]
- 25.322.4.17 `int vtkGDCMImageReader::LoadOverlays` [protected]
- 25.322.4.18 `int vtkGDCMImageReader::LossyFlag` [protected]
- 25.322.4.19 `vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties` [protected]
- 25.322.4.20 `int vtkGDCMImageReader::NumberOfIconImages` [protected]
- 25.322.4.21 `int vtkGDCMImageReader::NumberOfOverlays` [protected]

25.322.4.22 int vtkGDCMImageReader::PlanarConfiguration [protected]

25.322.4.23 double vtkGDCMImageReader::Scale [protected]

25.322.4.24 double vtkGDCMImageReader::Shift [protected]

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

- vtkGDCMImageReader.h

25.323 vtkGDCMImageWriter Class Reference

```
#include <vtkGDCMImageWriter.h>
```

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)

25.323.1 Detailed Description

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingle-BitTo8Bits.cxx, gdcmortoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, MagnifyFile.cxx, and RefCounting.cs.

25.323.2 Member Enumeration Documentation

25.323.2.1 enum vtkGDCMImageWriter::CompressionTypes

Enumerator:

NO_COMPRESSION
JPEG_COMPRESSION
JPEG2000_COMPRESSION
JPEGLS_COMPRESSION
RLE_COMPRESSION

25.323.3 Constructor & Destructor Documentation

25.323.3.1 vtkGDCMImageWriter::vtkGDCMImageWriter () [protected]

25.323.3.2 vtkGDCMImageWriter::~~vtkGDCMImageWriter () [protected]

25.323.4 Member Function Documentation

25.323.4.1 virtual const char* vtkGDCMImageWriter::GetDescriptiveName () [inline],[virtual]

25.323.4.2 virtual const char* vtkGDCMImageWriter::GetFileExtensions () [inline],[virtual]

25.323.4.3 virtual char* vtkGDCMImageWriter::GetFileName () [protected],[virtual]

25.323.4.4 static vtkGDCMImageWriter* vtkGDCMImageWriter::New () [static]

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorphoplanes.cxx, HelloActiviz.cs, HelloVTKWorld.cs, HelloVTKWorld2.cs, MagnifyFile.cxx, and RefCounting.cs.

25.323.4.5 virtual void vtkGDCMImageWriter::PrintSelf (ostream & os, vtkIndent indent) [virtual]

25.323.4.6 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.

25.323.4.7 virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (const double dircos[6]) [virtual]

25.323.4.8 virtual void vtkGDCMImageWriter::SetFileNames (vtkStringArray *) [virtual]

Examples:

ConvertMultiFrameToSingleFrame.cxx.

25.323.4.9 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.

25.323.4.10 vtkGDCMImageWriter::vtkBooleanMacro (LossyFlag , int)

25.323.4.11 vtkGDCMImageWriter::vtkBooleanMacro (FileLowerLeft , int)

25.323.4.12 vtkGDCMImageWriter::vtkGetMacro (LossyFlag , int)

25.323.4.13 vtkGDCMImageWriter::vtkGetMacro (Shift , double)

25.323.4.14 vtkGDCMImageWriter::vtkGetMacro (Scale , double)

- 25.323.4.15 `vtkGDCMImageWriter::vtkGetMacro (ImageFormat , int)`
- 25.323.4.16 `vtkGDCMImageWriter::vtkGetMacro (FileLowerLeft , int)`
- 25.323.4.17 `vtkGDCMImageWriter::vtkGetMacro (PlanarConfiguration , int)`
- 25.323.4.18 `vtkGDCMImageWriter::vtkGetMacro (CompressionType , int)`
- 25.323.4.19 `vtkGDCMImageWriter::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)`
- 25.323.4.20 `vtkGDCMImageWriter::vtkGetObjectMacro (FileNames , vtkStringArray)`
- 25.323.4.21 `vtkGDCMImageWriter::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)`
- 25.323.4.22 `vtkGDCMImageWriter::vtkGetStringMacro (StudyUID)`
- 25.323.4.23 `vtkGDCMImageWriter::vtkGetStringMacro (SeriesUID)`
- 25.323.4.24 `vtkGDCMImageWriter::vtkSetMacro (LossyFlag , int)`
- 25.323.4.25 `vtkGDCMImageWriter::vtkSetMacro (Shift , double)`
- 25.323.4.26 `vtkGDCMImageWriter::vtkSetMacro (Scale , double)`
- 25.323.4.27 `vtkGDCMImageWriter::vtkSetMacro (ImageFormat , int)`
- 25.323.4.28 `vtkGDCMImageWriter::vtkSetMacro (FileLowerLeft , int)`
- 25.323.4.29 `vtkGDCMImageWriter::vtkSetMacro (PlanarConfiguration , int)`
- 25.323.4.30 `vtkGDCMImageWriter::vtkSetMacro (CompressionType , int)`
- 25.323.4.31 `vtkGDCMImageWriter::vtkSetStringMacro (StudyUID)`
- 25.323.4.32 `vtkGDCMImageWriter::vtkSetStringMacro (SeriesUID)`
- 25.323.4.33 `vtkGDCMImageWriter::vtkTypeRevisionMacro (vtkGDCMImageWriter , vtkImageWriter)`
- 25.323.4.34 `virtual void vtkGDCMImageWriter::Write () [virtual]`

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingle-BitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, and MagnifyFile.cxx.

- 25.323.4.35 `int vtkGDCMImageWriter::WriteGDCMData (vtkImageData * data , int timeStep) [protected]`
- 25.323.4.36 `void vtkGDCMImageWriter::WriteSlice (vtkImageData * data) [protected]`

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

- vtkGDCMImageWriter.h

25.324 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

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 vtkGDCMImageWriter

25.324.1 Constructor & Destructor Documentation

25.324.1.1 vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties () [protected]

25.324.1.2 vtkGDCMMedicalImageProperties::~~vtkGDCMMedicalImageProperties () [protected]

25.324.2 Member Function Documentation

25.324.2.1 virtual void vtkGDCMMedicalImageProperties::Clear () [virtual]

25.324.2.2 gdcmm::File const& vtkGDCMMedicalImageProperties::GetFile (unsigned int t) [protected]

25.324.2.3 static vtkGDCMMedicalImageProperties* vtkGDCMMedicalImageProperties::New () [static]

25.324.2.4 void vtkGDCMMedicalImageProperties::PrintSelf (ostream & os, vtkIndent indent)

25.324.2.5 void vtkGDCMMedicalImageProperties::PushBackFile (gdcmm::File const & f) [protected]

25.324.2.6 `vtkGDCMMedicalImageProperties::vtkTypeRevisionMacro (vtkGDCMMedicalImageProperties ,
vtkMedicalImageProperties)`

25.324.3 Friends And Related Function Documentation

25.324.3.1 `friend class vtkGDCMImageReader` [`friend`]

25.324.3.2 `friend class vtkGDCMImageWriter` [`friend`]

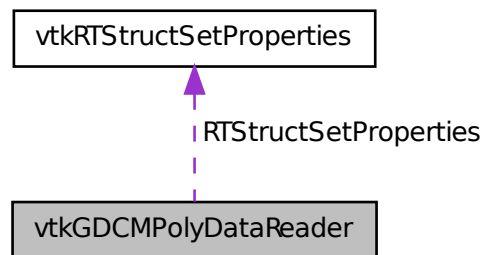
The documentation for this class was generated from the following file:

- `vtkGDCMMedicalImageProperties.h`

25.325 vtkGDCMPolyDataReader Class Reference

```
#include <vtkGDCMPolyDataReader.h>
```

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 gdcm::Reader &reader)`
- `int RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)`
- `int RequestData_HemodynamicWaveformStorage (gdcm::Reader const &reader, vtkInformationVector *outputVector)`
- `int RequestData_RTStructureSetStorage (gdcm::Reader const &reader, vtkInformationVector *outputVector)`
- `int RequestInformation (vtkInformation *vtkNotUsed(request), vtkInformationVector **vtkNotUsed(inputVector), vtkInformationVector *outputVector)`
- `int RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const &reader)`
- `int RequestInformation_RTStructureSetStorage (gdcm::Reader const &reader)`

Protected Attributes

- `char * FileName`
- `vtkMedicalImageProperties * MedicalImageProperties`
- `vtkRTStructSetProperties * RTStructSetProperties`

25.325.1 Detailed Description

Examples:

`gdcmscene.cxx`, `GenerateRTSTRUCT.cxx`, and `rtstructapp.cxx`.

25.325.2 Constructor & Destructor Documentation

25.325.2.1 `vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ()` `[protected]`

25.325.2.2 `vtkGDCMPolyDataReader::~~vtkGDCMPolyDataReader ()` `[protected]`

25.325.3 Member Function Documentation

25.325.3.1 `void vtkGDCMPolyDataReader::FillMedicalImageInformation (const gdcm::Reader & reader)` `[protected]`

25.325.3.2 `static vtkGDCMPolyDataReader* vtkGDCMPolyDataReader::New ()` `[static]`

Examples:

`gdcmscene.cxx`, `GenerateRTSTRUCT.cxx`, and `rtstructapp.cxx`.

25.325.3.3 `virtual void vtkGDCMPolyDataReader::PrintSelf (ostream & os, vtkIndent indent)` `[virtual]`

25.325.3.4 `int vtkGDCMPolyDataReader::RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)` `[protected]`

25.325.3.5 `int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (gdcm::Reader const & reader, vtkInformationVector * outputVector)` `[protected]`

- 25.325.3.6 `int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (gdcm::Reader const & reader, vtkInformationVector * outputVector)` [protected]
- 25.325.3.7 `int vtkGDCMPolyDataReader::RequestInformation (vtkInformation * vtkNotUsedrequest, vtkInformationVector ** vtkNotUsedinputVector, vtkInformationVector * outputVector)` [protected]
- 25.325.3.8 `int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const & reader)` [protected]
- 25.325.3.9 `int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (gdcm::Reader const & reader)` [protected]
- 25.325.3.10 `vtkGDCMPolyDataReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)`
- 25.325.3.11 `vtkGDCMPolyDataReader::vtkGetObjectMacro (RTStructSetProperties , vtkRTStructSetProperties)`
- 25.325.3.12 `vtkGDCMPolyDataReader::vtkGetStringMacro (FileName)`
- 25.325.3.13 `vtkGDCMPolyDataReader::vtkSetStringMacro (FileName)`
- 25.325.3.14 `vtkGDCMPolyDataReader::vtkTypeRevisionMacro (vtkGDCMPolyDataReader , vtkPolyDataAlgorithm)`

25.325.4 Member Data Documentation

- 25.325.4.1 `char* vtkGDCMPolyDataReader::FileName` [protected]
- 25.325.4.2 `vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties` [protected]
- 25.325.4.3 `vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties` [protected]

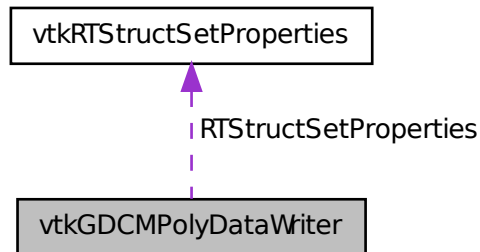
The documentation for this class was generated from the following file:

- `vtkGDCMPolyDataReader.h`

25.326 vtkGDCMPolyDataWriter Class Reference

```
#include <vtkGDCMPolyDataWriter.h>
```

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`

25.326.1 Detailed Description

Examples:

`GenerateRTSTRUCT.cxx`, and `rtstructapp.cxx`.

25.326.2 Constructor & Destructor Documentation

25.326.2.1 `vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ()` [protected]

25.326.2.2 `vtkGDCMPolyDataWriter::~~vtkGDCMPolyDataWriter ()` [protected]

25.326.3 Member Function Documentation

25.326.3.1 `void vtkGDCMPolyDataWriter::InitializeRTStructSet (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray * inROINames, vtkStringArray * inROIAlgorithmName, vtkStringArray * inROIType)`

Examples:

GenerateRTSTRUCT.cxx.

25.326.3.2 `static vtkGDCMPolyDataWriter* vtkGDCMPolyDataWriter::New ()` [static]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.3.3 `virtual void vtkGDCMPolyDataWriter::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

25.326.3.4 `virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (vtkMedicalImageProperties * pd)` [virtual]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.3.5 `void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (int n)`

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.3.6 `virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (vtkRTStructSetProperties * pd)` [virtual]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

25.326.3.7 `vtkGDCMPolyDataWriter::vtkTypeRevisionMacro (vtkGDCMPolyDataWriter , vtkPolyDataWriter)`

25.326.3.8 `void vtkGDCMPolyDataWriter::WriteData ()` [protected]

25.326.3.9 `void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (gdcm::File & file, int num)` [protected]

25.326.3.10 `void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (gdcm::File & file)` [protected]

25.326.4 Member Data Documentation

25.326.4.1 `vtkMedicalImageProperties*` `vtkGDCMPolyDataWriter::MedicalImageProperties` [protected]

25.326.4.2 `vtkRTStructSetProperties*` `vtkGDCMPolyDataWriter::RTStructSetProperties` [protected]

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

- `vtkGDCMPolyDataWriter.h`

25.327 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Public Types

- `typedef const char *const (* MD5MetalmagesType) [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 * GetMD5Metalmage (unsigned int file)`
- `static const char * GetMHDMD5FromFile (const char *filepath)`
- `static unsigned int GetNumberOfMD5Metalmages ()`
- `static const char * GetRAWMD5FromFile (const char *filepath)`
- `static const char * GetVTKDataRoot ()`
- `static vtkGDCMTesting * New ()`

Protected Member Functions

- `vtkGDCMTesting ()`
- `~vtkGDCMTesting ()`

25.327.1 Detailed Description

Examples:

HelloActiviz5.cs, HelloVTKWorld2.cs, MetalmageMD5Activiz.cs, ReadSeriesIntoVTK.java, and RefCounting.cs.

25.327.2 Member Typedef Documentation

25.327.2.1 `typedef const char* const(* vtkGDCMTesting::MD5MetalmagesType)[3]`

25.327.3 Constructor & Destructor Documentation

25.327.3.1 `vtkGDCMTesting::vtkGDCMTesting ()` [protected]

25.327.3.2 `vtkGDCMTesting::~~vtkGDCMTesting ()` [protected]

25.327.4 Member Function Documentation

25.327.4.1 `static const char* vtkGDCMTesting::GetGDCMDataRoot ()` [static]

Examples:

HelloActiviz5.cs, and ReadSeriesIntoVTK.java.

25.327.4.2 `static const char* const* vtkGDCMTesting::GetMD5Metalmage (unsigned int file)` [static]

25.327.4.3 `static const char* vtkGDCMTesting::GetMHDM5FromFile (const char * filepath)` [static]

Examples:

MetalmageMD5Activiz.cs.

25.327.4.4 `static unsigned int vtkGDCMTesting::GetNumberOfMD5Metalmages ()` [static]

25.327.4.5 `static const char* vtkGDCMTesting::GetRAWMD5FromFile (const char * filepath)` [static]

Examples:

MetalmageMD5Activiz.cs.

25.327.4.6 `static const char* vtkGDCMTesting::GetVTKDataRoot ()` [static]

Examples:

HelloActiviz5.cs, and HelloVTKWorld2.cs.

25.327.4.7 `static vtkGDCMTesting* vtkGDCMTesting::New ()` [static]

Examples:

RefCounting.cs.

25.327.4.8 void vtkGDCMTesting::PrintSelf (ostream & *os*, vtkIndent *indent*)

25.327.4.9 vtkGDCMTesting::vtkTypeRevisionMacro (vtkGDCMTesting , vtkObject)

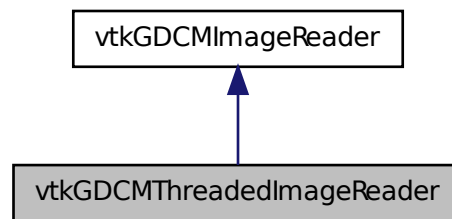
The documentation for this class was generated from the following file:

- vtkGDCMTesting.h

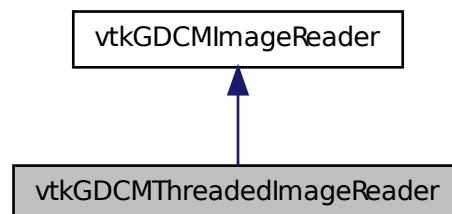
25.328 vtkGDCMThreadedImageReader Class Reference

```
#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

25.328.1 Constructor & Destructor Documentation

25.328.1.1 `vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ()` `[protected]`

25.328.1.2 `vtkGDCMThreadedImageReader::~~vtkGDCMThreadedImageReader ()` `[protected]`

25.328.2 Member Function Documentation

25.328.2.1 `void vtkGDCMThreadedImageReader::ExecuteData (vtkDataObject * out)` `[protected]`

Reimplemented from `vtkGDCMImageReader`.

25.328.2.2 `void vtkGDCMThreadedImageReader::ExecuteInformation ()` `[protected]`

Reimplemented from `vtkGDCMImageReader`.

25.328.2.3 `static vtkGDCMThreadedImageReader* vtkGDCMThreadedImageReader::New ()` `[static]`

Reimplemented from `vtkGDCMImageReader`.

25.328.2.4 `virtual void vtkGDCMThreadedImageReader::PrintSelf (ostream & os, vtkIndent indent)` `[virtual]`

Reimplemented from `vtkGDCMImageReader`.

25.328.2.5 void vtkGDCMThreadedImageReader::ReadFiles (unsigned int *nfiles*, const char * *filenames*[]) [protected]

25.328.2.6 void vtkGDCMThreadedImageReader::RequestDataCompat () [protected]

Reimplemented from vtkGDCMImageReader.

25.328.2.7 vtkGDCMThreadedImageReader::vtkBooleanMacro (UseShiftScale , int)

25.328.2.8 vtkGDCMThreadedImageReader::vtkGetMacro (UseShiftScale , int)

25.328.2.9 vtkGDCMThreadedImageReader::vtkSetMacro (Shift , double)

25.328.2.10 vtkGDCMThreadedImageReader::vtkSetMacro (Scale , double)

25.328.2.11 vtkGDCMThreadedImageReader::vtkSetMacro (UseShiftScale , int)

25.328.2.12 vtkGDCMThreadedImageReader::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader ,
vtkGDCMImageReader)

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

- vtkGDCMThreadedImageReader.h

25.329 vtkGDCMThreadedImageReader2 Class Reference

```
#include <vtkGDCMThreadedImageReader2.h>
```

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)`

25.329.1 Constructor & Destructor Documentation

25.329.1.1 `vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ()` [protected]

25.329.1.2 `vtkGDCMThreadedImageReader2::~~vtkGDCMThreadedImageReader2 ()` [protected]

25.329.2 Member Function Documentation

25.329.2.1 `virtual const char* vtkGDCMThreadedImageReader2::GetFileName (int i = 0)` [virtual]

25.329.2.2 `static vtkGDCMThreadedImageReader2* vtkGDCMThreadedImageReader2::New ()` [static]

25.329.2.3 `virtual void vtkGDCMThreadedImageReader2::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

25.329.2.4 `int vtkGDCMThreadedImageReader2::RequestInformation (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector)` [protected]

25.329.2.5 `virtual void vtkGDCMThreadedImageReader2::SetFileName (const char * filename)` [virtual]

25.329.2.6 `virtual void vtkGDCMThreadedImageReader2::SetFileNames (vtkStringArray *)` [virtual]

25.329.2.7 `int vtkGDCMThreadedImageReader2::SplitExtent (int splitExt[6], int startExt[6], int num, int total)`

- 25.329.2.8 void vtkGDCMThreadedImageReader2::ThreadedRequestData (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*, vtkImageData *** *inData*, vtkImageData ** *outData*, int *outExt*[6], int *id*) [protected]
- 25.329.2.9 vtkGDCMThreadedImageReader2::vtkBooleanMacro (FileLowerLeft , int)
- 25.329.2.10 vtkGDCMThreadedImageReader2::vtkBooleanMacro (LoadOverlays , int)
- 25.329.2.11 vtkGDCMThreadedImageReader2::vtkBooleanMacro (UseShiftScale , int)
- 25.329.2.12 vtkGDCMThreadedImageReader2::vtkGetMacro (FileLowerLeft , int)
- 25.329.2.13 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfOverlays , int)
- 25.329.2.14 vtkGDCMThreadedImageReader2::vtkGetMacro (DataScalarType , int)
- 25.329.2.15 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfScalarComponents , int)
- 25.329.2.16 vtkGDCMThreadedImageReader2::vtkGetMacro (LoadOverlays , int)
- 25.329.2.17 vtkGDCMThreadedImageReader2::vtkGetMacro (Shift , double)
- 25.329.2.18 vtkGDCMThreadedImageReader2::vtkGetMacro (Scale , double)
- 25.329.2.19 vtkGDCMThreadedImageReader2::vtkGetMacro (UseShiftScale , int)
- 25.329.2.20 vtkGDCMThreadedImageReader2::vtkGetObjectMacro (FileNames , vtkStringArray)
- 25.329.2.21 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataOrigin , double)
- 25.329.2.22 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataSpacing , double)
- 25.329.2.23 vtkGDCMThreadedImageReader2::vtkGetVector6Macro (DataExtent , int)
- 25.329.2.24 vtkGDCMThreadedImageReader2::vtkSetMacro (FileLowerLeft , int)
- 25.329.2.25 vtkGDCMThreadedImageReader2::vtkSetMacro (DataScalarType , int)
- 25.329.2.26 vtkGDCMThreadedImageReader2::vtkSetMacro (NumberOfScalarComponents , int)
- 25.329.2.27 vtkGDCMThreadedImageReader2::vtkSetMacro (LoadOverlays , int)
- 25.329.2.28 vtkGDCMThreadedImageReader2::vtkSetMacro (Shift , double)
- 25.329.2.29 vtkGDCMThreadedImageReader2::vtkSetMacro (Scale , double)
- 25.329.2.30 vtkGDCMThreadedImageReader2::vtkSetMacro (UseShiftScale , int)
- 25.329.2.31 vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataOrigin , double)
- 25.329.2.32 vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataSpacing , double)

25.329.2.33 `vtkGDCMThreadedImageReader2::vtkSetVector6Macro (DataExtent , int)`

25.329.2.34 `vtkGDCMThreadedImageReader2::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader2 ,
vtkThreadedImageAlgorithm)`

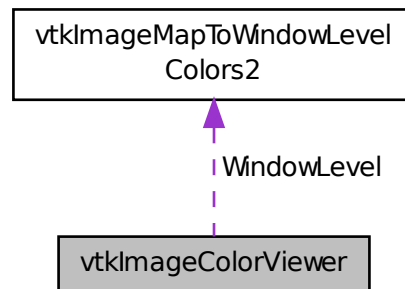
The documentation for this class was generated from the following file:

- `vtkGDCMThreadedImageReader2.h`

25.330 vtkImageColorViewer Class Reference

```
#include <vtkImageColorViewer.h>
```

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`

25.330.1 Detailed Description

Examples:

`gdcmrtionplan.cxx`, and `gdcmrtplan.cxx`.

25.330.2 Member Enumeration Documentation

25.330.2.1 anonymous enum

Enumerator:

SLICE_ORIENTATION_YZ
SLICE_ORIENTATION_XZ
SLICE_ORIENTATION_XY

25.330.3 Constructor & Destructor Documentation

25.330.3.1 `vtkImageColorViewer::vtkImageColorViewer ()` [protected]

25.330.3.2 `vtkImageColorViewer::~~vtkImageColorViewer ()` [protected]

25.330.4 Member Function Documentation

25.330.4.1 `virtual void vtkImageColorViewer::AddInput (vtkImageData * input)` [virtual]

25.330.4.2 `virtual void vtkImageColorViewer::AddInputConnection (vtkAlgorithmOutput * input)` [virtual]

25.330.4.3 `virtual double vtkImageColorViewer::GetColorLevel ()` [virtual]

- 25.330.4.4 virtual double vtkImageColorViewer::GetColorWindow () [virtual]
- 25.330.4.5 virtual vtkImageData* vtkImageColorViewer::GetInput () [virtual]
- 25.330.4.6 virtual int vtkImageColorViewer::GetOffScreenRendering () [virtual]
- 25.330.4.7 double vtkImageColorViewer::GetOverlayVisibility ()
- 25.330.4.8 virtual int* vtkImageColorViewer::GetPosition () [virtual]
- 25.330.4.9 virtual int* vtkImageColorViewer::GetSize () [virtual]
- 25.330.4.10 virtual int vtkImageColorViewer::GetSliceMax () [virtual]
- 25.330.4.11 virtual int vtkImageColorViewer::GetSliceMin () [virtual]
- 25.330.4.12 virtual void vtkImageColorViewer::GetSliceRange (int *range*[2]) [inline],[virtual]

References GetSliceRange().

Referenced by GetSliceRange().

- 25.330.4.13 virtual void vtkImageColorViewer::GetSliceRange (int & *min*, int & *max*) [virtual]
- 25.330.4.14 virtual int* vtkImageColorViewer::GetSliceRange () [virtual]
- 25.330.4.15 virtual const char* vtkImageColorViewer::GetWindowName () [virtual]
- 25.330.4.16 virtual void vtkImageColorViewer::InstallPipeline () [protected],[virtual]
- 25.330.4.17 static vtkImageColorViewer* vtkImageColorViewer::New () [static]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

- 25.330.4.18 void vtkImageColorViewer::PrintSelf (ostream & *os*, vtkIndent *indent*)
- 25.330.4.19 virtual void vtkImageColorViewer::Render (void) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

- 25.330.4.20 virtual void vtkImageColorViewer::SetColorLevel (double *s*) [virtual]
- 25.330.4.21 virtual void vtkImageColorViewer::SetColorWindow (double *s*) [virtual]
- 25.330.4.22 virtual void vtkImageColorViewer::SetDisplayId (void * *a*) [virtual]

25.330.4.23 `virtual void vtkImageColorViewer::SetInput (vtkImageData * in)` [virtual]

Examples:

`gdcmrtionplan.cxx`, and `gdcmrtplan.cxx`.

25.330.4.24 `virtual void vtkImageColorViewer::SetInputConnection (vtkAlgorithmOutput * input)` [virtual]

25.330.4.25 `virtual void vtkImageColorViewer::SetOffScreenRendering (int)` [virtual]

25.330.4.26 `void vtkImageColorViewer::SetOverlayVisibility (double vis)`

25.330.4.27 `virtual void vtkImageColorViewer::SetParentId (void * a)` [virtual]

25.330.4.28 `virtual void vtkImageColorViewer::SetPosition (int a, int b)` [virtual]

25.330.4.29 `virtual void vtkImageColorViewer::SetPosition (int a[2])` [inline],[virtual]

References `SetPosition()`.

Referenced by `SetPosition()`.

25.330.4.30 `virtual void vtkImageColorViewer::SetRenderer (vtkRenderer * arg)` [virtual]

25.330.4.31 `virtual void vtkImageColorViewer::SetRenderWindow (vtkRenderWindow * arg)` [virtual]

25.330.4.32 `virtual void vtkImageColorViewer::SetSize (int a, int b)` [virtual]

Examples:

`gdcmrtionplan.cxx`, and `gdcmrtplan.cxx`.

25.330.4.33 `virtual void vtkImageColorViewer::SetSize (int a[2])` [inline],[virtual]

References `SetSize()`.

Referenced by `SetSize()`.

25.330.4.34 `virtual void vtkImageColorViewer::SetSlice (int s)` [virtual]

25.330.4.35 `virtual void vtkImageColorViewer::SetSliceOrientation (int orientation)` [virtual]

25.330.4.36 `virtual void vtkImageColorViewer::SetSliceOrientationToXY ()` [inline],[virtual]

References `SLICE_ORIENTATION_XY`.

25.330.4.37 `virtual void vtkImageColorViewer::SetSliceOrientationToXZ ()` [inline],[virtual]

References `SLICE_ORIENTATION_XZ`.

25.330.4.38 virtual void vtkImageColorViewer::SetSliceOrientationToYZ () [inline],[virtual]

References SLICE_ORIENTATION_YZ.

25.330.4.39 virtual void vtkImageColorViewer::SetupInteractor (vtkRenderWindowInteractor *) [virtual]

Examples:

gdcmrtonplan.cxx, and gdcmrtpplan.cxx.

25.330.4.40 virtual void vtkImageColorViewer::SetWindowId (void * a) [virtual]

25.330.4.41 virtual void vtkImageColorViewer::UnInstallPipeline () [protected],[virtual]

25.330.4.42 virtual void vtkImageColorViewer::UpdateDisplayExtent () [virtual]

25.330.4.43 virtual void vtkImageColorViewer::UpdateOrientation () [protected],[virtual]

25.330.4.44 vtkImageColorViewer::VTK_LEGACY (int GetWholeZMin())

25.330.4.45 vtkImageColorViewer::VTK_LEGACY (int GetWholeZMax())

25.330.4.46 vtkImageColorViewer::VTK_LEGACY (int GetZSlice())

25.330.4.47 vtkImageColorViewer::VTK_LEGACY (void SetZSliceint)

25.330.4.48 vtkImageColorViewer::vtkBooleanMacro (OffScreenRendering , int)

25.330.4.49 vtkImageColorViewer::vtkGetMacro (SliceOrientation , int)

25.330.4.50 vtkImageColorViewer::vtkGetMacro (Slice , int)

25.330.4.51 vtkImageColorViewer::vtkGetObjectMacro (RenderWindow , vtkRenderWindow)

25.330.4.52 vtkImageColorViewer::vtkGetObjectMacro (Renderer , vtkRenderer)

25.330.4.53 vtkImageColorViewer::vtkGetObjectMacro (ImageActor , vtkImageActor)

25.330.4.54 vtkImageColorViewer::vtkGetObjectMacro (WindowLevel , vtkImageMapToWindowLevelColors2)

25.330.4.55 vtkImageColorViewer::vtkGetObjectMacro (InteractorStyle , vtkInteractorStyleImage)

25.330.4.56 vtkImageColorViewer::vtkTypeRevisionMacro (vtkImageColorViewer , vtkObject)

25.330.5 Member Data Documentation

25.330.5.1 int vtkImageColorViewer::FirstRender [protected]

25.330.5.2 vtkImageActor* vtkImageColorViewer::ImageActor [protected]

- 25.330.5.3 `vtkRenderWindowInteractor* vtkImageColorViewer::Interactor` [protected]
- 25.330.5.4 `vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle` [protected]
- 25.330.5.5 `vtkImageActor* vtkImageColorViewer::OverlayImageActor` [protected]
- 25.330.5.6 `vtkRenderer* vtkImageColorViewer::Renderer` [protected]
- 25.330.5.7 `vtkRenderWindow* vtkImageColorViewer::RenderWindow` [protected]
- 25.330.5.8 `int vtkImageColorViewer::Slice` [protected]
- 25.330.5.9 `int vtkImageColorViewer::SliceOrientation` [protected]
- 25.330.5.10 `vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel` [protected]

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

- `vtkImageColorViewer.h`

25.331 `vtkImageMapToColors16` Class Reference

```
#include <vtkImageMapToColors16.h>
```

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

25.331.1 Constructor & Destructor Documentation

25.331.1.1 vtkImageMapToColors16::vtkImageMapToColors16 () [protected]

25.331.1.2 vtkImageMapToColors16::~~vtkImageMapToColors16 () [protected]

25.331.2 Member Function Documentation

25.331.2.1 virtual unsigned long vtkImageMapToColors16::GetMTime () [virtual]

25.331.2.2 static vtkImageMapToColors16* vtkImageMapToColors16::New () [static]

25.331.2.3 void vtkImageMapToColors16::PrintSelf (ostream & os, vtkIndent indent)

25.331.2.4 virtual int vtkImageMapToColors16::RequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector) [protected],[virtual]

25.331.2.5 virtual int vtkImageMapToColors16::RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *) [protected],[virtual]

25.331.2.6 virtual void vtkImageMapToColors16::SetLookupTable (vtkScalarsToColors *) [virtual]

25.331.2.7 void vtkImageMapToColors16::SetOutputFormatToLuminance () [inline]

25.331.2.8 void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha () [inline]

25.331.2.9 void vtkImageMapToColors16::SetOutputFormatToRGB () [inline]

25.331.2.10 void vtkImageMapToColors16::SetOutputFormatToRGBA () [inline]

25.331.2.11 void vtkImageMapToColors16::ThreadedRequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector, vtkImageData *** inData, vtkImageData ** outData, int extent[6], int id) [protected]

- 25.331.2.12 `vtkImageMapToColors16::vtkBooleanMacro (PassAlphaToOutput , int)`
- 25.331.2.13 `vtkImageMapToColors16::vtkGetMacro (OutputFormat , int)`
- 25.331.2.14 `vtkImageMapToColors16::vtkGetMacro (ActiveComponent , int)`
- 25.331.2.15 `vtkImageMapToColors16::vtkGetMacro (PassAlphaToOutput , int)`
- 25.331.2.16 `vtkImageMapToColors16::vtkGetObjectMacro (LookupTable , vtkScalarsToColors)`
- 25.331.2.17 `vtkImageMapToColors16::vtkSetMacro (OutputFormat , int)`
- 25.331.2.18 `vtkImageMapToColors16::vtkSetMacro (ActiveComponent , int)`
- 25.331.2.19 `vtkImageMapToColors16::vtkSetMacro (PassAlphaToOutput , int)`
- 25.331.2.20 `vtkImageMapToColors16::vtkTypeRevisionMacro (vtkImageMapToColors16 , vtkThreadedImageAlgorithm)`

25.331.3 Member Data Documentation

- 25.331.3.1 `int vtkImageMapToColors16::ActiveComponent` [protected]
- 25.331.3.2 `int vtkImageMapToColors16::DataWasPassed` [protected]
- 25.331.3.3 `vtkScalarsToColors* vtkImageMapToColors16::LookupTable` [protected]
- 25.331.3.4 `int vtkImageMapToColors16::OutputFormat` [protected]
- 25.331.3.5 `int vtkImageMapToColors16::PassAlphaToOutput` [protected]

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

- `vtkImageMapToColors16.h`

25.332 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

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

25.332.1 Constructor & Destructor Documentation

25.332.1.1 vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 () [protected]

25.332.1.2 vtkImageMapToWindowLevelColors2::~~vtkImageMapToWindowLevelColors2 () [protected]

25.332.2 Member Function Documentation

25.332.2.1 static vtkImageMapToWindowLevelColors2* vtkImageMapToWindowLevelColors2::New () [static]

25.332.2.2 void vtkImageMapToWindowLevelColors2::PrintSelf (ostream & os, vtkIndent indent)

25.332.2.3 virtual int vtkImageMapToWindowLevelColors2::RequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector) [protected], [virtual]

25.332.2.4 virtual int vtkImageMapToWindowLevelColors2::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected], [virtual]

25.332.2.5 void vtkImageMapToWindowLevelColors2::ThreadedRequestData (vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector, vtkImageData *** inData, vtkImageData ** outData, int extent[6], int id) [protected]

25.332.2.6 vtkImageMapToWindowLevelColors2::vtkGetMacro (Window , double)

25.332.2.7 vtkImageMapToWindowLevelColors2::vtkGetMacro (Level , double)

25.332.2.8 vtkImageMapToWindowLevelColors2::vtkSetMacro (Window , double)

25.332.2.9 vtkImageMapToWindowLevelColors2::vtkSetMacro (Level , double)

25.332.2.10 `vtkImageMapToWindowLevelColors2::vtkTypeRevisionMacro (vtkImageMapToWindowLevelColors2 ,
vtkImageMapToColors)`

25.332.3 Member Data Documentation

25.332.3.1 `double vtkImageMapToWindowLevelColors2::Level` `[protected]`

25.332.3.2 `double vtkImageMapToWindowLevelColors2::Window` `[protected]`

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

- `vtkImageMapToWindowLevelColors2.h`

25.333 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

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 *)`

25.333.1 Constructor & Destructor Documentation

25.333.1.1 `vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ()` `[protected]`

25.333.1.2 `vtkImagePlanarComponentsToComponents::~~vtkImagePlanarComponentsToComponents ()` `[inline],
[protected]`

25.333.2 Member Function Documentation

25.333.2.1 `static vtkImagePlanarComponentsToComponents* vtkImagePlanarComponentsToComponents::New ()`
`[static]`

25.333.2.2 `void vtkImagePlanarComponentsToComponents::PrintSelf (ostream & os, vtkIndent indent)`

- 25.333.2.3 `virtual int vtkImagePlanarComponentsToComponents::RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *) [protected],[virtual]`
- 25.333.2.4 `vtkImagePlanarComponentsToComponents::vtkTypeRevisionMacro (vtkImagePlanarComponentsToComponents , vtkImageAlgorithm)`

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

- `vtkImagePlanarComponentsToComponents.h`

25.334 vtkImageRGBToYBR Class Reference

```
#include <vtkImageRGBToYBR.h>
```

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)`

25.334.1 Constructor & Destructor Documentation

- 25.334.1.1 `vtkImageRGBToYBR::vtkImageRGBToYBR () [protected]`
- 25.334.1.2 `vtkImageRGBToYBR::~~vtkImageRGBToYBR () [inline],[protected]`

25.334.2 Member Function Documentation

- 25.334.2.1 `static vtkImageRGBToYBR* vtkImageRGBToYBR::New () [static]`
- 25.334.2.2 `void vtkImageRGBToYBR::PrintSelf (ostream & os, vtkIndent indent)`
- 25.334.2.3 `void vtkImageRGBToYBR::ThreadedExecute (vtkImageData * inData, vtkImageData * outData, int ext[6], int id) [protected]`
- 25.334.2.4 `vtkImageRGBToYBR::vtkTypeRevisionMacro (vtkImageRGBToYBR , vtkThreadedImageAlgorithm)`

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

- `vtkImageRGBToYBR.h`

25.335 vtkImageYBRToRGB Class Reference

```
#include <vtkImageYBRToRGB.h>
```

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)

25.335.1 Constructor & Destructor Documentation

25.335.1.1 `vtkImageYBRToRGB::vtkImageYBRToRGB ()` [protected]

25.335.1.2 `vtkImageYBRToRGB::~~vtkImageYBRToRGB ()` [inline],[protected]

25.335.2 Member Function Documentation

25.335.2.1 `static vtkImageYBRToRGB* vtkImageYBRToRGB::New ()` [static]

25.335.2.2 `void vtkImageYBRToRGB::PrintSelf (ostream & os, vtkIndent indent)`

25.335.2.3 `void vtkImageYBRToRGB::ThreadedExecute (vtkImageData * inData, vtkImageData * outData, int ext[6], int id)`
[protected]

25.335.2.4 `vtkImageYBRToRGB::vtkTypeRevisionMacro (vtkImageYBRToRGB , vtkThreadedImageAlgorithm)`

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

- vtkImageYBRToRGB.h

25.336 vtkLookupTable16 Class Reference

```
#include <vtkLookupTable16.h>
```

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

25.336.1 Constructor & Destructor Documentation

25.336.1.1 `vtkLookupTable16::vtkLookupTable16 (int size = 256, int ext = 256)` [protected]

25.336.1.2 `vtkLookupTable16::~~vtkLookupTable16 ()` [protected]

25.336.2 Member Function Documentation

25.336.2.1 `void vtkLookupTable16::Build ()`

25.336.2.2 `unsigned short* vtkLookupTable16::GetPointer (const vtkIdType id)` [inline]

25.336.2.3 `void vtkLookupTable16::MapScalarsThroughTable2 (void * input, unsigned char * output, int inputDataType, int numberOfValues, int inputIncrement, int outputFormat)` [protected]

25.336.2.4 `static vtkLookupTable16* vtkLookupTable16::New ()` [static]

25.336.2.5 `void vtkLookupTable16::PrintSelf (ostream & os, vtkIndent indent)`

25.336.2.6 `void vtkLookupTable16::SetNumberOfTableValues (vtkIdType number)`

25.336.2.7 `vtkLookupTable16::vtkTypeRevisionMacro (vtkLookupTable16 , vtkLookupTable)`

25.336.2.8 `unsigned char * vtkLookupTable16::WritePointer (const vtkIdType id, const int number)` [inline]

References Table16.

25.336.3 Member Data Documentation

25.336.3.1 vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]

Referenced by WritePointer().

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

- vtkLookupTable16.h

25.337 vtkRTStructSetProperties Class Reference

```
#include <vtkRTStructSetProperties.h>
```

Public Member Functions

- void AddContourReferencedFrameOfReference (vtkIdType pdnum, const char *classuid, const char *instanceuid)
- void AddReferencedFrameOfReference (const char *classuid, const char *instanceuid)
- void AddStructureSetROI (int roinumber, const char *refframerefid, const char *roiname, const char *ROI-GenerationAlgorithm)
- void AddStructureSetROIObservation (int refnumber, int observationnumber, const char *rtroiinterpretedtype, const char *roiinterpreter)
- 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 * GetStructureSetROIGenerationAlgorithm (vtkIdType)
- const char * GetStructureSetROIName (vtkIdType)
- int GetStructureSetROINumber (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

25.337.1 Detailed Description

Examples:

```
GenerateRTSTRUCT.cxx.
```

25.337.2 Constructor & Destructor Documentation

25.337.2.1 `vtkRTStructSetProperties::vtkRTStructSetProperties ()` [protected]

25.337.2.2 `vtkRTStructSetProperties::~~vtkRTStructSetProperties ()` [protected]

25.337.3 Member Function Documentation

25.337.3.1 `void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (vtkIdType pdnum, const char * classuid, const char * instanceuid)`

25.337.3.2 `void vtkRTStructSetProperties::AddReferencedFrameOfReference (const char * classuid, const char * instanceuid)`

- 25.337.3.3 void vtkRTStructSetProperties::AddStructureSetROI (int *roinumber*, const char * *refframerefid*, const char * *roiname*, const char * *ROIGenerationAlgorithm*)
- 25.337.3.4 void vtkRTStructSetProperties::AddStructureSetROIObservation (int *refnumber*, int *observationnumber*, const char * *rtroiinterpretedtype*, const char * *roiinterpreter*)
- 25.337.3.5 virtual void vtkRTStructSetProperties::Clear () [virtual]
- 25.337.3.6 virtual void vtkRTStructSetProperties::DeepCopy (vtkRTStructSetProperties * *p*) [virtual]
- 25.337.3.7 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (vtkIdType *pdnum*, vtkIdType *id*)
- 25.337.3.8 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType *pdnum*, vtkIdType *id*)
- 25.337.3.9 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ()
- 25.337.3.10 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (vtkIdType *pdnum*)
- 25.337.3.11 vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ()
- 25.337.3.12 vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ()
- 25.337.3.13 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (vtkIdType *id*)
- 25.337.3.14 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (vtkIdType *id*)
- 25.337.3.15 int vtkRTStructSetProperties::GetStructureSetObservationNumber (vtkIdType *id*)
- 25.337.3.16 const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (vtkIdType)
- 25.337.3.17 const char* vtkRTStructSetProperties::GetStructureSetROIName (vtkIdType)
- 25.337.3.18 int vtkRTStructSetProperties::GetStructureSetROINumber (vtkIdType *id*)
- 25.337.3.19 const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (vtkIdType)
- 25.337.3.20 const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (vtkIdType *id*)
- 25.337.3.21 static vtkRTStructSetProperties* vtkRTStructSetProperties::New () [static]

Examples:

GenerateRTSTRUCT.cxx.

- 25.337.3.22 void vtkRTStructSetProperties::PrintSelf (ostream & *os*, vtkIndent *indent*)
- 25.337.3.23 vtkRTStructSetProperties::vtkGetStringMacro (StructureSetLabel)
- 25.337.3.24 vtkRTStructSetProperties::vtkGetStringMacro (StructureSetName)

- 25.337.3.25 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetDate)`
- 25.337.3.26 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetTime)`
- 25.337.3.27 `vtkRTStructSetProperties::vtkGetStringMacro (SOPInstanceUID)`
- 25.337.3.28 `vtkRTStructSetProperties::vtkGetStringMacro (StudyInstanceUID)`
- 25.337.3.29 `vtkRTStructSetProperties::vtkGetStringMacro (SeriesInstanceUID)`
- 25.337.3.30 `vtkRTStructSetProperties::vtkGetStringMacro (ReferenceSeriesInstanceUID)`
- 25.337.3.31 `vtkRTStructSetProperties::vtkGetStringMacro (ReferenceFrameOfReferenceUID)`
- 25.337.3.32 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetLabel)`
- 25.337.3.33 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetName)`
- 25.337.3.34 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetDate)`
- 25.337.3.35 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetTime)`
- 25.337.3.36 `vtkRTStructSetProperties::vtkSetStringMacro (SOPInstanceUID)`
- 25.337.3.37 `vtkRTStructSetProperties::vtkSetStringMacro (StudyInstanceUID)`
- 25.337.3.38 `vtkRTStructSetProperties::vtkSetStringMacro (SeriesInstanceUID)`
- 25.337.3.39 `vtkRTStructSetProperties::vtkSetStringMacro (ReferenceSeriesInstanceUID)`
- 25.337.3.40 `vtkRTStructSetProperties::vtkSetStringMacro (ReferenceFrameOfReferenceUID)`
- 25.337.3.41 `vtkRTStructSetProperties::vtkTypeRevisionMacro (vtkRTStructSetProperties , vtkObject)`

25.337.4 Member Data Documentation

- 25.337.4.1 `vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals` [protected]
- 25.337.4.2 `char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID` [protected]
- 25.337.4.3 `char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID` [protected]
- 25.337.4.4 `char* vtkRTStructSetProperties::SeriesInstanceUID` [protected]
- 25.337.4.5 `char* vtkRTStructSetProperties::SOPInstanceUID` [protected]
- 25.337.4.6 `char* vtkRTStructSetProperties::StructureSetDate` [protected]
- 25.337.4.7 `char* vtkRTStructSetProperties::StructureSetLabel` [protected]
- 25.337.4.8 `char* vtkRTStructSetProperties::StructureSetName` [protected]

25.337.4.9 `char* vtkRTStructSetProperties::StructureSetTime` [protected]

25.337.4.10 `char* vtkRTStructSetProperties::StudyInstanceUID` [protected]

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

- `vtkRTStructSetProperties.h`

25.338 `gdcm::Waveform` Class Reference

Waveform class.

```
#include <gdcmWaveform.h>
```

Public Member Functions

- `Waveform ()`

25.338.1 Detailed Description

Waveform class.

25.338.2 Constructor & Destructor Documentation

25.338.2.1 `gdcm::Waveform::Waveform ()` [inline]

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

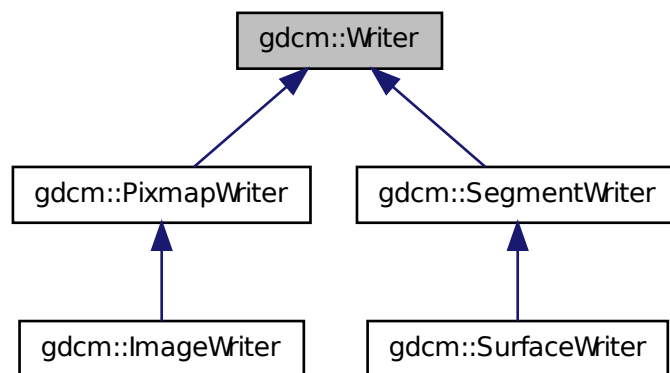
- `gdcmWaveform.h`

25.339 `gdcm::Writer` Class Reference

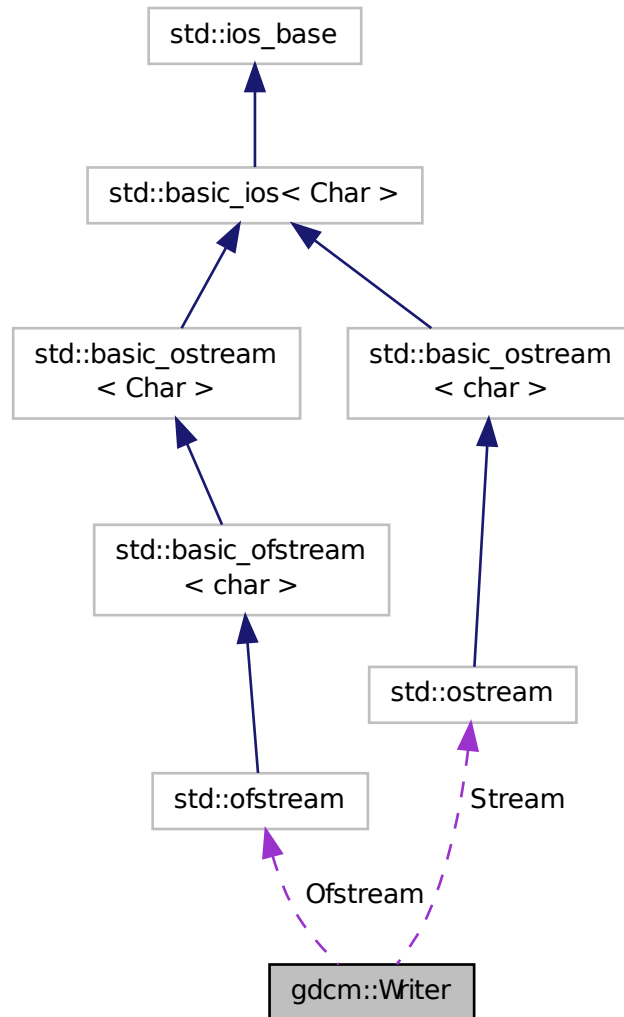
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.

```
#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

25.339.1 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:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

25.339.2 Constructor & Destructor Documentation

25.339.2.1 `gdcm::Writer::Writer ()`

25.339.2.2 `virtual gdcm::Writer::~~Writer ()` `[virtual]`

25.339.3 Member Function Documentation

25.339.3.1 `void gdcm::Writer::CheckFileMetaInformationOff ()` `[inline]`

Examples:

FixBrokenJ2K.cxx, and HelloWorld.cxx.

25.339.3.2 `void gdcm::Writer::CheckFileMetaInformationOn ()` `[inline]`

25.339.3.3 `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, iU22tomultisc.cxx, pmsct_rgb1.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

25.339.3.4 `std::ostream* gdcm::Writer::GetStreamPtr () const` `[inline]`, `[protected]`

25.339.3.5 `void gdcm::Writer::SetCheckFileMetaInformation (bool b)` `[inline]`

Undocumented function, do not use (= leave default)

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and PatchFile.cxx.

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

Set/Get the DICOM file (DataSet + Header)

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, DuplicatePCDE.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakelImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.339.3.7 void gdcm::Writer::SetFileName (const char * *filename_native*)

Set the filename of DICOM file to write:

Examples:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakelIdentifyFile.cxx, GenFakelImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloVizWorld.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.339.3.8 void gdcm::Writer::SetStream (std::ostream & *output_stream*) [inline]

Set user ostream buffer.

25.339.3.9 void gdcm::Writer::SetWriteDataSetOnly (bool *b*) [inline],[protected]

25.339.3.10 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:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakelIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

25.339.4 Friends And Related Function Documentation

25.339.4.1 friend class StreamImageWriter [friend]

25.339.5 Member Data Documentation

25.339.5.1 std::ofstream* gdcm::Writer::Ofstream [protected]

25.339.5.2 std::ostream* gdcm::Writer::Stream [protected]

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

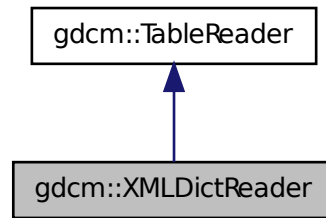
- gdcmWriter.h

25.340 gdcM::XMLDictReader Class Reference

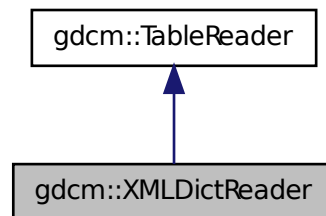
Class for representing a XMLDictReader.

```
#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)

25.340.1 Detailed Description

Class for representing a XMLDictReader.

Note

bla Will read the DICOMV3.xml file

25.340.2 Constructor & Destructor Documentation

25.340.2.1 gdcm::XMLDictReader::XMLDictReader ()

25.340.2.2 gdcm::XMLDictReader::~~XMLDictReader () [inline]

25.340.3 Member Function Documentation

25.340.3.1 void gdcm::XMLDictReader::CharacterDataHandler (const char * *data*, int *length*) [virtual]

Reimplemented from gdcm::TableReader.

25.340.3.2 void gdcm::XMLDictReader::EndElement (const char * *name*) [virtual]

Reimplemented from gdcm::TableReader.

25.340.3.3 const Dict& gdcm::XMLDictReader::GetDict () [inline]

25.340.3.4 void gdcm::XMLDictReader::HandleDescription (const char ** *atts*) [protected]

25.340.3.5 void gdcm::XMLDictReader::HandleEntry (const char ** *atts*) [protected]

25.340.3.6 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:

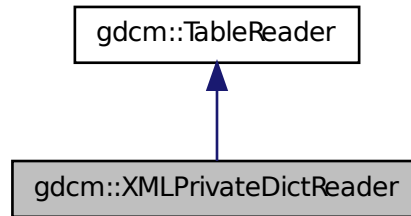
- gdcmXMLDictReader.h

25.341 gdcm::XMLPrivateDictReader Class Reference

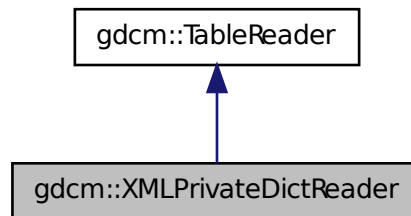
Class for representing a XMLPrivateDictReader.

```
#include <gdcmXMLPrivateDictReader.h>
```

Inheritance diagram for `gdc::XMLPrivateDictReader`:



Collaboration diagram for `gdc::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)`

25.341.1 Detailed Description

Class for representing a `XMLPrivateDictReader`.

Note

bla Will read the Private.xml file

25.341.2 Constructor & Destructor Documentation

25.341.2.1 `gdcm::XMLPrivateDictReader::XMLPrivateDictReader ()`

25.341.2.2 `gdcm::XMLPrivateDictReader::~~XMLPrivateDictReader ()` `[inline]`

25.341.3 Member Function Documentation

25.341.3.1 `void gdcm::XMLPrivateDictReader::CharacterDataHandler (const char * data, int length)` `[virtual]`

Reimplemented from `gdcm::TableReader`.

25.341.3.2 `void gdcm::XMLPrivateDictReader::EndElement (const char * name)` `[virtual]`

Reimplemented from `gdcm::TableReader`.

25.341.3.3 `const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ()` `[inline]`

25.341.3.4 `void gdcm::XMLPrivateDictReader::HandleDescription (const char ** atts)` `[protected]`

25.341.3.5 `void gdcm::XMLPrivateDictReader::HandleEntry (const char ** atts)` `[protected]`

25.341.3.6 `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 26

File Documentation

26.1 gdc2pnm.man File Reference

26.2 gdc2vtk.man File Reference

26.3 gdcAAabortPDU.h File Reference

Classes

- class gdc::network::AAabortPDU
AAabortPDU Table 9-26 A-ABORT PDU FIELDS.

Namespaces

- namespace gdc
- namespace gdc::network

26.4 gdcAAAssociateACPDU.h File Reference

Classes

- class gdc::network::AAAssociateACPDU
AAAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

Namespaces

- namespace gdc
- namespace gdc::network

26.5 gdcmAAssociateRJPDU.h File Reference

Classes

- class gdcm::network::AAssociateRJPDU
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.6 gdcmAAssociateRQPDU.h File Reference

Classes

- class gdcm::network::AAssociateRQPDU
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.7 gdcmAbstractSyntax.h File Reference

Classes

- class gdcm::network::AbstractSyntax
AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.8 gdcmanon.man File Reference

26.9 gdcmAnonymizeEvent.h File Reference

Classes

- class gdcm::AnonymizeEvent
AnonymizeEvent Special type of event triggered during the Anonymization process.

Namespaces

- namespace gdcm

26.10 gdcmAnonymizer.h File Reference

Classes

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

Namespaces

- namespace gdcm

26.11 gdcmApplicationContext.h File Reference

Classes

- class gdcm::network::ApplicationContext
ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.12 gdcmApplicationEntity.h File Reference

Classes

- class gdcm::ApplicationEntity
ApplicationEntity.

Namespaces

- namespace gdcm

26.13 gdcmAReleaseRPPDU.h File Reference

Classes

- class gdcm::network::AReleaseRPPDU
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

Namespaces

- namespace gdcn
- namespace gdcn::network

26.14 gdcnAReleaseRQPDU.h File Reference

Classes

- class gdcn::network::AReleaseRQPDU
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

Namespaces

- namespace gdcn
- namespace gdcn::network

26.15 gdcnARTIMTimer.h File Reference

Classes

- class gdcn::network::ARTIMTimer
ARTIMTimer This file contains the code for the ARTIM timer.

Namespaces

- namespace gdcn
- namespace gdcn::network

26.16 gdcnASN1.h File Reference

Classes

- class gdcn::ASN1
Class for ASN1.

Namespaces

- namespace gdcn

26.17 gdcmAsynchronousOperationsWindowSub.h File Reference

Classes

- class gdcm::network::AsynchronousOperationsWindowSub
AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.18 gdcmAttribute.h File Reference

Classes

- class gdcm::Attribute< Group, Element, TVR, TVM >
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.
- class gdcm::Attribute< Group, Element, TVR, VM::VM1 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM2_n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM3_n >
- class gdcm::VRVLSize< 0 >
- class gdcm::VRVLSize< 1 >

Namespaces

- namespace gdcm

26.19 gdcmAudioCodec.h File Reference

Classes

- class gdcm::AudioCodec
AudioCodec.

Namespaces

- namespace gdcm

26.20 gdcmBase64.h File Reference

Classes

- class `gdcm::Base64`
Class for Base64.

Namespaces

- namespace `gdcm`

26.21 gdcmBaseCompositeMessage.h File Reference

Classes

- class `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.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

26.22 gdcmBasePDU.h File Reference

Classes

- class `gdcm::network::BasePDU`
BasePDU base class for PDUs.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

26.23 gdcmBaseRootQuery.h File Reference

Classes

- class `gdcm::BaseRootQuery`

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::EQueryLevel {
gdcm::ePatient,
gdcm::eStudy,
gdcm::eSeries,
gdcm::eImageOrFrame }

BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root name and date: 18 oct 2010 mmr.

- enum gdcm::EQueryType {
gdcm::eFind,
gdcm::eMove }

26.24 gdcmBasicOffsetTable.h File Reference

Classes

- class gdcm::BasicOffsetTable
Class to represent a BasicOffsetTable.

Namespaces

- namespace gdcm

Functions

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

26.25 gdcmBitmap.h File Reference

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)

Namespaces

- namespace gdcm

26.26 gdcmBitmapToBitmapFilter.h File Reference

Classes

- class gdcm::BitmapToBitmapFilter
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

Namespaces

- namespace gdcm

26.27 gdcmByteBuffer.h File Reference

Classes

- class gdcm::ByteBuffer
ByteBuffer.

Namespaces

- namespace gdcm

26.28 gdcmByteSwap.h File Reference

Classes

- class gdcm::ByteSwap< T >
ByteSwap.

Namespaces

- namespace gdcm

26.29 gdcmByteSwapFilter.h File Reference

Classes

- class gdcm::ByteSwapFilter
ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

Namespaces

- namespace gdcm

26.30 gdcmByteValue.h File Reference

Classes

- class gdcm::ByteValue
Class to represent binary value (array of bytes)

Namespaces

- namespace gdcm

26.31 gdcmCEchoMessages.h File Reference

Classes

- class gdcm::network::CEchoRQ
CEchoRQ this file defines the messages for the echo action.
- class gdcm::network::CEchoRSP

Namespaces

- namespace gdcm
- namespace gdcm::network

26.32 gdcmCFindMessages.h File Reference

Classes

- class gdcm::network::CFindCancelRQ
- class gdcm::network::CFindRQ
- class gdcm::network::CFindRSP

Namespaces

- namespace gdcm
- namespace gdcm::network

26.33 gdcmCMoveMessages.h File Reference

Classes

- 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.

Namespaces

- namespace gdcmm
- namespace gdcmm::network

26.34 gdcmmCodec.h File Reference

Classes

- class gdcmm::Codec
Codec class.

Namespaces

- namespace gdcmm

26.35 gdcmmCoder.h File Reference

Classes

- class gdcmm::Coder
Coder.

Namespaces

- namespace gdcmm

26.36 gdcmmCodeString.h File Reference

Classes

- class gdcmm::CodeString
CodeString This is an implementation of DICOM VR: CS The ctor will properly Trim so that operator== is correct.

Namespaces

- namespace gdcmm

Functions

- bool gdcmm::operator!= (const CodeString &ref, const CodeString &cs)
- std::ostream & gdcmm::operator<< (std::ostream &os, const CodeString &str)
- bool gdcmm::operator== (const CodeString &ref, const CodeString &cs)

26.37 gdcmCommand.h File Reference

Classes

- class `gdcm::Command`
Command superclass for callback/observer methods.
- class `gdcm::MemberCommand< T >`
Command subclass that calls a pointer to a member function.
- class `gdcm::SimpleMemberCommand< T >`
Command subclass that calls a pointer to a member function.

Namespaces

- namespace `gdcm`

26.38 gdcmCommandDataSet.h File Reference

Classes

- class `gdcm::CommandDataSet`
Class to represent a Command DataSet.

Namespaces

- namespace `gdcm`

Functions

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

26.39 gdcmCompositeMessageFactory.h File Reference

Classes

- class `gdcm::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).

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

26.40 gdcmCompositeNetworkFunctions.h File Reference

Classes

- class gdcm::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:

Namespaces

- namespace gdcm

26.41 gdcmConstCharWrapper.h File Reference

Classes

- class gdcm::ConstCharWrapper
Do not use me.

Namespaces

- namespace gdcm

26.42 gdcmconv.man File Reference

26.43 gdcmCP246ExplicitDataElement.h File Reference

Classes

- class gdcm::CP246ExplicitDataElement
Class to read/write a DataElement as CP246Explicit Data Element.

Namespaces

- namespace gdcm

26.44 gdcmCryptographicMessageSyntax.h File Reference

Classes

- class gdcm::CryptographicMessageSyntax

Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

Namespaces

- namespace gdcm

26.45 gdcmCSAElement.h File Reference

Classes

- class gdcm::CSAElement
Class to represent a CSA Element.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)

26.46 gdcmCSAHeader.h File Reference

Classes

- class gdcm::CSAHeader
Class for CSAHeader.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)

26.47 gdcmCSAHeaderDict.h File Reference

Classes

- class gdcm::CSAHeaderDict
Class to represent a map of CSAHeaderDictEntry.
- class gdcm::CSAHeaderDictException

Namespaces

- namespace gdcM

Functions

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

26.48 gdcMCSAHeaderDictEntry.h File Reference

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

- namespace gdcM

Functions

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

26.49 gdcMCStoreMessages.h File Reference

Classes

- class `gdcM::network::CStoreRQ`
CStoreRQ this file defines the messages for the echo action.
- class `gdcM::network::CStoreRSP`

Namespaces

- namespace gdcM
- namespace `gdcM::network`

26.50 gdcMCurve.h File Reference

Classes

- class `gdcM::Curve`
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

Namespaces

- namespace gdcm

26.51 gdcmDataElement.h File Reference

Classes

- class gdcm::DataElement
Class to represent a Data Element either Implicit or Explicit.

Namespaces

- namespace gdcm

Functions

- bool gdcm::operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & gdcm::operator<< (std::ostream &os, const DataElement &val)

26.52 gdcmDataEvent.h File Reference

Classes

- class gdcm::DataEvent
DataEvent.

Namespaces

- namespace gdcm

26.53 gdcmDataSet.h File Reference

Classes

- class gdcm::DataElementException
- class gdcm::DataSet
Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object.

Namespaces

- namespace gdcm

Functions

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

26.54 gdcmmDataSetEvent.h File Reference

Classes

- class `gdcmm::DataSetEvent`
DataSetEvent Special type of event triggered during the DataSet store/move process.

Namespaces

- namespace `gdcmm`

26.55 gdcmmDataSetHelper.h File Reference

Classes

- class `gdcmm::DataSetHelper`
DataSetHelper (internal class, not intended for user level)

Namespaces

- namespace `gdcmm`

26.56 gdcmmDecoder.h File Reference

Classes

- class `gdcmm::Decoder`
Decoder.

Namespaces

- namespace `gdcmm`

26.57 gdcmmDefinedTerms.h File Reference

Classes

- class `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.

Namespaces

- namespace gdcm

26.58 gdcmDeflateStream.h File Reference

26.59 gdcmDefs.h File Reference

Classes

- class gdcm::Defs

FIXME I do not like the name 'Defs'.

Namespaces

- namespace gdcm

26.60 gdcmDeltaEncodingCodec.h File Reference

Classes

- class gdcm::DeltaEncodingCodec

DeltaEncodingCodec compression used by some private vendor.

Namespaces

- namespace gdcm

26.61 gdcmDICOMDIR.h File Reference

Classes

- class gdcm::DICOMDIR

DICOMDIR class.

Namespaces

- namespace gdcm

26.62 gdcmDICOmdirGenerator.h File Reference

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

- namespace gdcm

26.63 gdcmDict.h File Reference

Classes

- class gdcm::Dict
Class to represent a map of DictEntry.
- class gdcm::PrivateDict
Private Dict.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)
- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)

26.64 gdcmDictConverter.h File Reference

Classes

- class gdcm::DictConverter
Class to convert a .dic file into something else:

Namespaces

- namespace gdcm

26.65 gdcmDictEntry.h File Reference

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

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)

26.66 gdcmDictPrinter.h File Reference

Classes

- class gdcm::DictPrinter

DictPrinter class.

Namespaces

- namespace gdcm

26.67 gdcmDicts.h File Reference

Classes

- class gdcm::Dicts

Class to manipulate the sum of knowledge (all the dict user load)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)

26.68 gdcmdiff.man File Reference

26.69 gdcmDIMSE.h File Reference

Classes

- class `gdcm::network::CEchoRQ`
CEchoRQ this file defines the messages for the cecho action.
- class `gdcm::network::CEchoRSP`
- 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

- namespace `gdcm`
- namespace `gdcm::network`

26.70 gdcmDirectionCosines.h File Reference

Classes

- class `gdcm::DirectionCosines`
class to handle DirectionCosines

Namespaces

- namespace `gdcm`

26.71 gdcmDirectory.h File Reference

Classes

- class `gdcm::Directory`
Class for manipulation directories.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Directory &d)`

26.72 gdcmDirectoryHelper.h File Reference

Classes

- class gdcm::DirectoryHelper

Namespaces

- namespace gdcm

26.73 gdcmDummyValueGenerator.h File Reference

Classes

- class gdcm::DummyValueGenerator
Class for generating dummy value.

Namespaces

- namespace gdcm

26.74 gdcmdump.man File Reference

26.75 gdcmDumper.h File Reference

Classes

- class gdcm::Dumper
Codec class.

Namespaces

- namespace gdcm

26.76 gdcmElement.h File Reference

Classes

- class gdcm::Element< TVR, TVM >
Element class.
- class gdcm::Element< TVR, VM::VM1_2 >
- class gdcm::Element< TVR, VM::VM1_n >
- class gdcm::Element< TVR, VM::VM2_2n >
- class gdcm::Element< TVR, VM::VM2_n >

- class gdcm::Element< TVR, VM::VM3_3n >
- class gdcm::Element< TVR, VM::VM3_n >
- class gdcm::Element< VR::AS, VM::VM5 >
- class gdcm::Element< VR::OB, VM::VM1 >
- class gdcm::Element< VR::OW, VM::VM1 >
- class gdcm::EncodingImplementation< VR::VRASCII >
- class gdcm::EncodingImplementation< VR::VRBINARY >
- struct gdcm::ignore_char

Namespaces

- namespace gdcm

Functions

- ignore_char const gdcm::backslash ('\\')
- std::istream & gdcm::operator>> (std::istream &in, ignore_char const &ic)
- template<typename Float >
std::string gdcm::to_string (Float data)

26.77 gdcmEncapsulatedDocument.h File Reference

Classes

- class gdcm::EncapsulatedDocument
EncapsulatedDocument.

Namespaces

- namespace gdcm

26.78 gdcmEnumeratedValues.h File Reference

Classes

- class gdcm::EnumeratedValues
Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

Namespaces

- namespace gdcm

26.79 gdcmEvent.h File Reference

Classes

- class gdcm::AbortEvent
- class gdcm::AnyEvent
- class gdcm::EndEvent
- class gdcm::Event
 - superclass for callback/observer methods*
- class gdcm::ExitEvent
- class gdcm::InitializeEvent
- class gdcm::IterationEvent
- class gdcm::ModifiedEvent
- class gdcm::NoEvent
- class gdcm::StartEvent
- class gdcm::UserEvent

Namespaces

- namespace gdcm

Macros

- #define gdcmEventMacro(classname, super)

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, Event &e)
 - Generic inserter operator for Event and its subclasses.*

26.79.1 Macro Definition Documentation

26.79.1.1 #define gdcmEventMacro(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 ::gdcm::Event* e) const \
    { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const \
    { return new Self; } \
    classname(const Self&s) : super(s){}; \
private: \
    void operator=(const Self&); \
}
```

26.80 gdcException.h File Reference

Classes

- class gdc::Exception
Exception.

Namespaces

- namespace gdc

26.81 gdcExplicitDataElement.h File Reference

Classes

- class gdc::ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element.

Namespaces

- namespace gdc

26.82 gdcExplicitImplicitDataElement.h File Reference

Classes

- class gdc::ExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element.

Namespaces

- namespace gdc

26.83 gdcFiducials.h File Reference

Classes

- class gdc::Fiducials
Fiducials.

Namespaces

- namespace gdc

26.84 gdcmFile.h File Reference

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

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const File &val)

26.85 gdcmFileDerivation.h File Reference

Classes

- class gdcm::FileDerivation

FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.

Namespaces

- namespace gdcm

26.86 gdcmFileExplicitFilter.h File Reference

Classes

- class gdcm::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.

Namespaces

- namespace gdcm

26.87 gdcmFileMetaInformation.h File Reference

Classes

- class `gdcm::FileMetaInformation`
Class to represent a File Meta Information.

Namespaces

- namespace `gdcm`

Functions

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

26.88 gdcmFilename.h File Reference

Classes

- class `gdcm::Filename`
Class to manipulate file name's.

Namespaces

- namespace `gdcm`

26.89 gdcmFilenameGenerator.h File Reference

Classes

- class `gdcm::FilenameGenerator`
FilenameGenerator.

Namespaces

- namespace `gdcm`

26.90 gdcmFileSet.h File Reference

Classes

- class `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.

Namespaces

- namespace gdcm

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)`

26.91 gdcmFindPatientRootQuery.h File Reference

Classes

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

Namespaces

- namespace gdcm

26.92 gdcmFindStudyRootQuery.h File Reference

Classes

- class `gdcm::FindStudyRootQuery`
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

Namespaces

- namespace gdcm

26.93 gdcmFragment.h File Reference

Classes

- class `gdcm::Fragment`
Class to represent a Fragment.

Namespaces

- namespace gdcm

Functions

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

26.94 gdcmgendir.man File Reference

26.95 gdcmGlobal.h File Reference

Classes

- class gdcm::Global
Global.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)

Variables

- static Global gdcm::GlobalInstance

26.96 gdcmGroupDict.h File Reference

Classes

- class gdcm::GroupDict
Class to represent the mapping from group number to its abbreviation and name.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)

26.97 gdcmlconImage.h File Reference

Namespaces

- namespace gdcm

Typedefs

- typedef Bitmap gdcmlconImage

26.98 gdcmlconImageFilter.h File Reference

Classes

- class gdcmlconImageFilter

IconImageFilter This filter will extract icons from a gdcml::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.

Namespaces

- namespace gdcml

26.99 gdcmlconImageGenerator.h File Reference

Classes

- class gdcmlconImageGenerator

IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of gdcml::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

Namespaces

- namespace gdcml

26.100 gdcmlImage.h File Reference

Classes

- class gdcmlImage

Image.

Namespaces

- namespace gdcml

26.101 gdcmImageApplyLookupTable.h File Reference

Classes

- class `gdcm::ImageApplyLookupTable`
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

Namespaces

- namespace `gdcm`

26.102 gdcmImageChangePhotometricInterpretation.h File Reference

Classes

- class `gdcm::ImageChangePhotometricInterpretation`
ImageChangePhotometricInterpretation class Class to change the Photometric Interpretation of an input DICOM.

Namespaces

- namespace `gdcm`

26.103 gdcmImageChangePlanarConfiguration.h File Reference

Classes

- class `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.

Namespaces

- namespace `gdcm`

26.104 gdcmImageChangeTransferSyntax.h File Reference

Classes

- class `gdcm::ImageChangeTransferSyntax`
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.

Namespaces

- namespace `gdcm`

26.105 gdcmImageCodec.h File Reference

Classes

- class gdcm::ImageCodec
ImageCodec.

Namespaces

- namespace gdcm

26.106 gdcmImageConverter.h File Reference

Classes

- class gdcm::ImageConverter
Image Converter.

Namespaces

- namespace gdcm

26.107 gdcmImageFragmentSplitter.h File Reference

Classes

- class gdcm::ImageFragmentSplitter
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

Namespaces

- namespace gdcm

26.108 gdcmImageHelper.h File Reference

Classes

- class gdcm::ImageHelper
ImageHelper (internal class, not intended for user level)

Namespaces

- namespace gdcm

26.109 gdcmImageReader.h File Reference

Classes

- class gdcm::ImageReader
ImageReader.

Namespaces

- namespace gdcm

26.110 gdcmImageToImageFilter.h File Reference

Classes

- class gdcm::ImageToImageFilter
ImageToImageFilter class Super class for all filter taking an image and producing an output image.

Namespaces

- namespace gdcm

26.111 gdcmImageWriter.h File Reference

Classes

- class gdcm::ImageWriter
ImageWriter.

Namespaces

- namespace gdcm

26.112 gdcming.man File Reference

26.113 gdcmImplementationClassUIDSub.h File Reference

Classes

- class gdcm::network::ImplementationClassUIDSub
ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE--RQ)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.114 gdcmImplementationUIDSub.h File Reference

Classes

- class gdcm::network::ImplementationUIDSub

ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.115 gdcmImplementationVersionNameSub.h File Reference

Classes

- class gdcm::network::ImplementationVersionNameSub

ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.116 gdcmImplicitDataElement.h File Reference

Classes

- class gdcm::ImplicitDataElement

Class to represent an Implicit VR Data Element.

Namespaces

- namespace gdcm

26.117 gdcminfo.man File Reference

26.118 gdcmIOD.h File Reference

Classes

- class `gdcm::IOD`
Class for representing a IOD.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)`

26.119 gdcmIODEntry.h File Reference

Classes

- class `gdcm::IODEntry`
Class for representing a IODEntry.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)`

26.120 gdcmIODs.h File Reference

Classes

- class `gdcm::IODs`
Class for representing a IODs.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)`

26.121 gdcmIPPSorter.h File Reference

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

- namespace `gdcm`

26.122 gdcmItem.h File Reference

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 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.

Namespaces

- namespace `gdcm`

Functions

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

26.123 gdcmJPEG12Codec.h File Reference

Classes

- class `gdcm::JPEG12Codec`

Class to do JPEG 12bits (lossy & lossless)

Namespaces

- namespace `gdcm`

26.124 gdcmJPEG16Codec.h File Reference

Classes

- class gdcm::JPEG16Codec
Class to do JPEG 16bits (lossless)

Namespaces

- namespace gdcm

26.125 gdcmJPEG2000Codec.h File Reference

Classes

- class gdcm::JPEG2000Codec
Class to do JPEG 2000.

Namespaces

- namespace gdcm

26.126 gdcmJPEG8Codec.h File Reference

Classes

- class gdcm::JPEG8Codec
Class to do JPEG 8bits (lossy & lossless)

Namespaces

- namespace gdcm

26.127 gdcmJPEGCodec.h File Reference

Classes

- class gdcm::JPEGCodec
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: gdcm::JPEG8Codec, gdcm::JPEG12Codec & gdcm::JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.

Namespaces

- namespace gdcm

26.128 gdcmJPEGLSCodec.h File Reference

Classes

- class gdcm::JPEGLSCodec
JPEG-LS.

Namespaces

- namespace gdcm

26.129 gdcmKAKADUCodec.h File Reference

Classes

- class gdcm::KAKADUCodec
KAKADUCodec.

Namespaces

- namespace gdcm

26.130 gdcmLegacyMacro.h File Reference

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.")`

26.130.1 Macro Definition Documentation

26.130.1.1 `#define GDCM_LEGACY(method) method;`

26.130.1.2 `#define GDCM_LEGACY_BODY(method, version) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")`

26.130.1.3 `#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.")`

26.131 gdcmLO.h File Reference

Classes

- class gdcm::LO

LO.

Namespaces

- namespace gdcM

26.132 gdcMLookupTable.h File Reference

Classes

- class gdcM::LookupTable
LookupTable class.

Namespaces

- namespace gdcM

26.133 gdcMacro.h File Reference

Classes

- class gdcM::Macro
Class for representing a Macro.

Namespaces

- namespace gdcM

Functions

- std::ostream & gdcM::operator<< (std::ostream &_os, const Macro &_val)

26.134 gdcMacroEntry.h File Reference

Macros

- #define GDCMMACROENTRY_H

26.134.1 Macro Definition Documentation

26.134.1.1 #define GDCMMACROENTRY_H

26.135 gdcMacros.h File Reference

Classes

- class gdcm::Macros
Class for representing a Modules.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)

26.136 gdcmMaximumLengthSub.h File Reference

Classes

- class gdcm::network::MaximumLengthSub
MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

- namespace gdcm
- namespace gdcm::network

26.137 gdcmMD5.h File Reference

Classes

- class gdcm::MD5
Class for MD5.

Namespaces

- namespace gdcm

26.138 gdcmMediaStorage.h File Reference

Classes

- class gdcm::MediaStorage
MediaStorage.

Namespaces

- namespace gdcM

Functions

- `std::ostream & gdcM::operator<< (std::ostream &_os, const MediaStorage &ms)`

26.139 gdcMMeshPrimitive.h File Reference

Classes

- class `gdcM::MeshPrimitive`

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

Namespaces

- namespace gdcM

26.140 gdcMModule.h File Reference

Classes

- class `gdcM::Module`

Class for representing a Module.

Namespaces

- namespace gdcM

Functions

- `std::ostream & gdcM::operator<< (std::ostream &_os, const Module &_val)`

26.141 gdcMModuleEntry.h File Reference

Classes

- class `gdcM::ModuleEntry`

Class for representing a ModuleEntry.

Namespaces

- namespace gdcM

Typedefs

- typedef ModuleEntry gdcm::MacroEntry

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)

26.142 gdcmModules.h File Reference

Classes

- class gdcm::Modules
Class for representing a Modules.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)

26.143 gdcmMovePatientRootQuery.h File Reference

Classes

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

Namespaces

- namespace gdcm

26.144 gdcmMoveStudyRootQuery.h File Reference

Classes

- class gdcm::MoveStudyRootQuery
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

Namespaces

- namespace gdcm

26.145 gdcmNestedModuleEntries.h File Reference

Classes

- class `gdcm::NestedModuleEntries`
Class for representing a NestedModuleEntries.

Namespaces

- namespace `gdcm`

Typedefs

- typedef `NestedModuleEntries` `gdcm::NestedMacroEntries`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const NestedModuleEntries &_val)`

26.146 gdcmNetworkEvents.h File Reference

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

Enumerations

- enum `gdcm::network::EEventID` {
 `gdcm::network::eAASSOCIATERequestLocalUser = 0`,
 `gdcm::network::eTransportConnConfirmLocal`,
 `gdcm::network::eASSOCIATE_ACPDUreceived`,
 `gdcm::network::eASSOCIATE_RJPDUreceived`,
 `gdcm::network::eTransportConnIndicLocal`,
 `gdcm::network::eAASSOCIATE_RQPDUreceived`,
 `gdcm::network::eAASSOCIATEResponseAccept`,
 `gdcm::network::eAASSOCIATEResponseReject`,
 `gdcm::network::ePDATArequest`,
 `gdcm::network::ePDATATFPDU`,
 `gdcm::network::eARELEASERequest`,
 `gdcm::network::eARELEASE_RQPDUReceivedOpen`,
 `gdcm::network::eARELEASE_RPPDUReceived`,
 `gdcm::network::eARELEASEResponse`,
 `gdcm::network::eAABORTRequest`,
 `gdcm::network::eAABORTPDUReceivedOpen`,
 `gdcm::network::eTransportConnectionClosed`,
 `gdcm::network::eARTIMTimerExpired`,
 `gdcm::network::eUnrecognizedPDUReceived`,
 `gdcm::network::eEventDoesNotExist` }

Variables

- `const int gdcm::network::cMaxEventID = eEventDoesNotExist`

26.147 gdcmNetworkStateID.h File Reference

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

Enumerations

- `enum gdcm::network::EStateID {
 gdcm::network::eStaDoesNotExist = 0,
 gdcm::network::eSta1Idle = 1,
 gdcm::network::eSta2Open = 2,
 gdcm::network::eSta3WaitLocalAssoc = 4,
 gdcm::network::eSta4LocalAssocDone = 8,
 gdcm::network::eSta5WaitRemoteAssoc = 16,
 gdcm::network::eSta6TransferReady = 32,
 gdcm::network::eSta7WaitRelease = 64,
 gdcm::network::eSta8WaitLocalRelease = 128,
 gdcm::network::eSta9ReleaseCollisionRqLocal = 256,
 gdcm::network::eSta10ReleaseCollisionAc = 512,
 gdcm::network::eSta11ReleaseCollisionRq = 1024,
 gdcm::network::eSta12ReleaseCollisionAcLocal = 2048,
 gdcm::network::eSta13AwaitingClose = 4096 }`

Functions

- `int gdcm::network::GetStateIndex (EStateID inState)`

Variables

- `const int gdcm::network::cMaxStateID = 13`

26.148 gdcmObject.h File Reference

Classes

- class `gdcm::Object`
Object.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Object &obj)`

26.149 gdcmmOrientation.h File Reference

Classes

- class `gdcmm::Orientation`
class to handle Orientation

Namespaces

- namespace `gdcmm`

Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Orientation &o)`

26.150 gdcmmOverlay.h File Reference

Classes

- class `gdcmm::Overlay`
Overlay class.

Namespaces

- namespace `gdcmm`

26.151 gdcmmParseException.h File Reference

Classes

- class `gdcmm::ParseException`
ParseException Standard exception handling object.

Namespaces

- namespace `gdcmm`

26.152 gdcmParser.h File Reference

Classes

- class gdcm::Parser
Parser ala XML_Parser from expat (SAX)

Namespaces

- namespace gdcm

26.153 gdcmPatient.h File Reference

Classes

- class gdcm::Patient
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

Namespaces

- namespace gdcm

26.154 gdcmPDataTFPDU.h File Reference

Classes

- class gdcm::network::PDataTFPDU
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.155 gdcmPDBElement.h File Reference

Classes

- class gdcm::PDBElement
Class to represent a PDB Element.

Namespaces

- namespace gdcm

Functions

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

26.156 gdcmmPDBHeader.h File Reference

Classes

- class `gdcmm::PDBHeader`
Class for PDBHeader.

Namespaces

- namespace `gdcmm`

Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const PDBHeader &d)`

26.157 gdcmmpdf.man File Reference

26.158 gdcmmPDFCodec.h File Reference

Classes

- class `gdcmm::PDFCodec`
PDFCodec class.

Namespaces

- namespace `gdcmm`

26.159 gdcmmPDUFactory.h File Reference

Classes

- class `gdcmm::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

- namespace `gdcmm`
- namespace `gdcmm::network`

26.160 gdcmPersonName.h File Reference

Classes

- class gdcm::PersonName
PersonName class.

Namespaces

- namespace gdcm

26.161 gdcmPhotometricInterpretation.h File Reference

Classes

- class gdcm::PhotometricInterpretation
Class to represent an PhotometricInterpretation.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)

26.162 gdcmPixelFormat.h File Reference

Classes

- class gdcm::PixelFormat
PixelFormat.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)

26.163 gdcmPixmap.h File Reference

Classes

- class 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)

Namespaces

- namespace gdcm

26.164 gdcmPixmapReader.h File Reference

Classes

- class gdcm::PixmapReader

PixmapReader.

Namespaces

- namespace gdcm

26.165 gdcmPixmapToPixmapFilter.h File Reference

Classes

- class gdcm::PixmapToPixmapFilter

PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.

Namespaces

- namespace gdcm

26.166 gdcmPixmapWriter.h File Reference

Classes

- class gdcm::PixmapWriter

PixmapWriter This class will takes two inputs:

Namespaces

- namespace gdcm

26.167 gdcmPNMCodec.h File Reference

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

- namespace gdcm

26.168 gdcmPreamble.h File Reference

Classes

- class gdcm::Preamble

DICOM Preamble (Part 10)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)

26.169 gdcmPresentationContext.h File Reference

Classes

- class gdcm::PresentationContext

PresentationContext.

Namespaces

- namespace gdcm

26.170 gdcmPresentationContextAC.h File Reference

Classes

- class gdcm::network::PresentationContextAC

PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

Namespaces

- namespace gdcM
- namespace gdcM::network

26.171 gdcMPresentationContextGenerator.h File Reference

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

- namespace gdcM

26.172 gdcMPresentationContextRQ.h File Reference

Classes

- class gdcM::network::PresentationContextRQ

PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

Namespaces

- namespace gdcM
- namespace gdcM::network

26.173 gdcMPresentationDataValue.h File Reference

Classes

- class gdcM::network::PresentationDataValue

PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

Namespaces

- namespace gdcM
- namespace gdcM::network

26.174 gdcmPrinter.h File Reference

Classes

- class gdcm::Printer
Printer class.

Namespaces

- namespace gdcm

26.175 gdcmPrivateTag.h File Reference

Classes

- class gdcm::PrivateTag
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)

26.176 gdcmProgressEvent.h File Reference

Classes

- class gdcm::ProgressEvent
ProgressEvent Special type of event triggered during.

Namespaces

- namespace gdcm

26.177 gdcmPVRGCodec.h File Reference

Classes

- class gdcm::PVRGCodec
PVRGCodec.

Namespaces

- namespace gdcm

26.178 gdcmPythonFilter.h File Reference

Classes

- class gdcm::PythonFilter

PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

Namespaces

- namespace gdcm

26.179 gdcmQueryBase.h File Reference

Classes

- class gdcm::QueryBase

QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ERootType {
gdcm::ePatientRootType,
gdcm::eStudyRootType }

26.180 gdcmQueryFactory.h File Reference

Classes

- class gdcm::QueryFactory

QueryFactory.h.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ECharSet {
gdcm::eLatin1 = 0,
gdcm::eLatin2,
gdcm::eLatin3,
gdcm::eLatin4,
gdcm::eCyrillic,
gdcm::eArabic,
gdcm::eGreek,
gdcm::eHebrew,
gdcm::eLatin5,
gdcm::eJapanese,
gdcm::eThai,
gdcm::eJapaneseKanjiMultibyte,
gdcm::eJapaneseSupplementaryKanjiMultibyte,
gdcm::eKoreanHangulHanjaMultibyte,
gdcm::eUTF8,
gdcm::eGB18030 }

26.181 gdcmQueryImage.h File Reference

Classes

- class gdcm::QueryImage
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

Namespaces

- namespace gdcm

26.182 gdcmQueryPatient.h File Reference

Classes

- class gdcm::QueryPatient
QueryPatient contains: class to construct a patient-based query for c-find and c-move.

Namespaces

- namespace gdcm

26.183 gdcmQuerySeries.h File Reference

Classes

- class gdcm::QuerySeries

QuerySeries contains: class to construct a series-based query for c-find and c-move.

Namespaces

- namespace gdcm

26.184 gdcmQueryStudy.h File Reference

Classes

- class gdcm::QueryStudy

QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

Namespaces

- namespace gdcm

26.185 gdcmraw.man File Reference

26.186 gdcmRAWCodec.h File Reference

Classes

- class gdcm::RAWCodec

RAWCodec class.

Namespaces

- namespace gdcm

26.187 gdcmReader.h File Reference

Classes

- class gdcm::Reader

Reader ala DOM (Document Object Model)

Namespaces

- namespace gdcm

26.188 gdcmRescaler.h File Reference

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

- namespace gdcm

26.189 gdcmRLECodec.h File Reference

Classes

- class gdcm::RLECodec

Class to do RLE.

Namespaces

- namespace gdcm

26.190 gdcmScanner.h File Reference

Classes

- struct gdcm::Scanner::ltstr
- class 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.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)

26.191 gdcmscanner.man File Reference

26.192 gdcmscu.man File Reference

26.193 gdcmSegment.h File Reference

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

- namespace gdcm

26.194 gdcmSegmentedPaletteColorLookupTable.h File Reference

Classes

- class gdcm::SegmentedPaletteColorLookupTable
SegmentedPaletteColorLookupTable class.

Namespaces

- namespace gdcm

26.195 gdcmSegmentHelper.h File Reference

Classes

- struct gdcm::SegmentHelper::BasicCodedEntry
This structure defines a basic coded entry with all of its attributes.

Namespaces

- namespace gdcm
- namespace gdcm::SegmentHelper

26.196 gdcmSegmentReader.h File Reference

Classes

- class gdcm::SegmentReader
This class defines a segment reader. It reads attributes of group 0x0062.

Namespaces

- namespace gdcm

26.197 gdcmSegmentWriter.h File Reference

Classes

- class gdcm::SegmentWriter
This class defines a segment writer. It writes attributes of group 0x0062.

Namespaces

- namespace gdcm

26.198 gdcmSequenceOfFragments.h File Reference

Classes

- class gdcm::SequenceOfFragments
Class to represent a Sequence Of Fragments.

Namespaces

- namespace gdcm

26.199 gdcmSequenceOfItems.h File Reference

Classes

- class gdcm::SequenceOfItems
Class to represent a Sequence Of Items (value representation : SQ)

Namespaces

- namespace gdcm

26.200 gdcmSerieHelper.h File Reference

Classes

- class gdcm::FileWithName
FileWithName.
- struct gdcm::SerieHelper::Rule
- class gdcm::SerieHelper

Namespaces

- namespace gdcM

Typedefs

- typedef bool(* gdcM::BOOL_FUNCTION_PFILE_PFILE_POINTER)(File *, File *)
- typedef std::vector<SmartPointer< FileWithName > > gdcM::FileList

Enumerations

- enum gdcM::CompOperators {
gdcM::GDCM_EQUAL = 0,
gdcM::GDCM_DIFFERENT,
gdcM::GDCM_GREATER,
gdcM::GDCM_GREATEROREQUAL,
gdcM::GDCM_LESS,
gdcM::GDCM_LESSEOREQUAL }
- enum gdcM::LodModeType {
gdcM::LD_ALL = 0x00000000,
gdcM::LD_NOSEQ = 0x00000001,
gdcM::LD_NOSHADOW = 0x00000002,
gdcM::LD_NOSHADOWSEQ = 0x00000004 }

26.201 gdcMSeries.h File Reference

Classes

- class gdcM::Series
Series.

Namespaces

- namespace gdcM

26.202 gdcMServiceClassUser.h File Reference

Classes

- class gdcM::ServiceClassUser
ServiceClassUser.

Namespaces

- namespace gdcM
- namespace gdcM::network

26.203 gdcmSHA1.h File Reference

Classes

- class gdcm::SHA1
Class for SHA1.

Namespaces

- namespace gdcm

26.204 gdcmSimpleSubjectWatcher.h File Reference

Classes

- class gdcm::SimpleSubjectWatcher
SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events.

Namespaces

- namespace gdcm

26.205 gdcmSmartPointer.h File Reference

Classes

- class gdcm::SmartPointer< ObjectType >
Class for Smart Pointer.

Namespaces

- namespace gdcm

26.206 gdcmSOPClassUIDToIOD.h File Reference

Classes

- class gdcm::SOPClassUIDToIOD
Class convert a class SOP Class UID into IOD.

Namespaces

- namespace gdcm

26.207 gdcmSorter.h File Reference

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

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)`

26.208 gdcmSpacing.h File Reference

Classes

- class `gdcm::Spacing`
Class for Spacing.

Namespaces

- namespace `gdcm`

26.209 gdcmSpectroscopy.h File Reference

Classes

- class `gdcm::Spectroscopy`
Spectroscopy class.

Namespaces

- namespace `gdcm`

26.210 gdcmSplitMosaicFilter.h File Reference

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

- namespace gdcm

26.211 gdcmStaticAssert.h File Reference

Classes

- struct gdcm::static_assert_test< x >
- struct gdcm::STATIC_ASSERTION_FAILURE< true >

Namespaces

- namespace gdcm

Macros

- #define GDCM_DO_JOIN(X, Y) GDCM_DO_JOIN2(X,Y)
- #define GDCM_DO_JOIN2(X, Y) X##Y
- #define GDCM_JOIN(X, Y) GDCM_DO_JOIN(X, Y)
- #define GDCM_STATIC_ASSERT(B)
*The GDCM_JOIN + **LINE** is needed to create a uniq identifier.*

26.211.1 Macro Definition Documentation

26.211.1.1 #define GDCM_DO_JOIN(X, Y) GDCM_DO_JOIN2(X,Y)

26.211.1.2 #define GDCM_DO_JOIN2(X, Y) X##Y

26.211.1.3 #define GDCM_JOIN(X, Y) GDCM_DO_JOIN(X, Y)

26.211.1.4 #define GDCM_STATIC_ASSERT(B)

Value:

```
typedef ::gdcm::static_assert_test<\
    sizeof( :gdcm::STATIC_ASSERTION_FAILURE< (bool) ( B ) > )>\
    GDCM_JOIN( gdcm_static_assert_typedef_, __LINE__ )
```

The GDCM_JOIN + **LINE** is needed to create a uniq identifier.

26.212 gdcmStreamImageReader.h File Reference

Classes

- struct gdcm::OneShotReadBuf
- class gdcm::StreamImageReader
StreamImageReader.

Namespaces

- namespace gdcm

26.213 gdcmStreamImageWriter.h File Reference

Classes

- class gdcm::StreamImageWriter
StreamImageReader.

Namespaces

- namespace gdcm

26.214 gdcmString.h File Reference

Classes

- class gdcm::String< TDelimiter, TMaxLength, TPadChar >
String.

Namespaces

- namespace gdcm

Functions

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

26.215 gdcmStringFilter.h File Reference

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

- namespace gdcm

26.216 gdcmStudy.h File Reference

Classes

- class gdcm::Study
Study.

Namespaces

- namespace gdcm

26.217 gdcmSubject.h File Reference

Classes

- class gdcm::Subject
Subject.

Namespaces

- namespace gdcm

26.218 gdcmSurface.h File Reference

Classes

- class gdcm::Surface
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

Namespaces

- namespace gdcm

26.219 gdcmSurfaceHelper.h File Reference

Classes

- class gdcm::SurfaceHelper

Namespaces

- namespace gdcm

26.220 gdcmSurfaceReader.h File Reference

Classes

- class gdcm::SurfaceReader

This class defines a SURFACE IE reader. It reads surface mesh module attributes.

Namespaces

- namespace gdcm

26.221 gdcmSurfaceWriter.h File Reference

Classes

- class gdcm::SurfaceWriter

This class defines a SURFACE IE writer. It writes surface mesh module attributes.

Namespaces

- namespace gdcm

26.222 gdcmSwapCode.h File Reference

Classes

- class gdcm::SwapCode

SwapCode representation.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)

26.223 gdcmSwapper.h File Reference

Classes

- class gdcm::SwapperDoOp
- class gdcm::SwapperNoOp

Namespaces

- namespace gdcm

26.224 gdcmSystem.h File Reference

Classes

- class gdcm::System
Class to do system operation.

Namespaces

- namespace gdcm

26.225 gdcmTable.h File Reference

Classes

- class gdcm::Table
Table.

Namespaces

- namespace gdcm

26.226 gdcmTableEntry.h File Reference

Classes

- class gdcm::TableEntry
TableEntry.

Namespaces

- namespace gdcm

26.227 gdcmTableReader.h File Reference

Classes

- class gdcm::TableReader
Class for representing a TableReader.

Namespaces

- namespace gdcM

26.228 gdcMTag.h File Reference

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

- namespace gdcM

Functions

- std::ostream & gdcM::operator<< (std::ostream &_os, const Tag &_val)
- std::istream & gdcM::operator>> (std::istream &_is, Tag &_val)

26.229 gdcMTagPath.h File Reference

Classes

- class gdcM::TagPath

class to handle a path of tag.

Namespaces

- namespace gdcM

26.230 gdcMTagToVR.h File Reference

Namespaces

- namespace gdcM

Functions

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

26.231 gdcmtar.man File Reference

26.232 gdcmTerminal.h File Reference

Namespaces

- namespace gdcmtar
- namespace gdcmtar::terminal

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

Enumerations

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

Functions

- GDCM_EXPORT std::string gdcmtar::terminal::setattribute (Attribute att)
- GDCM_EXPORT std::string gdcmtar::terminal::setbgcolor (Color c)
- GDCM_EXPORT std::string gdcmtar::terminal::setfgcolor (Color c)
- GDCM_EXPORT void gdcmtar::terminal::setmode (Mode m)

26.233 gdcmTestDriver.h File Reference

26.234 gdcmTesting.h File Reference

Classes

- class gdcmtar::Testing
class for testing

Namespaces

- namespace gdcm

26.235 gdcmTrace.h File Reference

Classes

- class gdcm::Trace
Trace.

Namespaces

- namespace gdcm

Macros

- #define GDCM_FUNCTION "<unknown>"
- #define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)
AssertAlways.
- #define gdcmAssertMacro(arg)
Assert.
- #define gdcmDebugMacro(msg)
Debug.
- #define gdcmErrorMacro(msg)
Error this is pretty bad, more than just warning It could mean lost of data, something not handle...
- #define gdcmWarningMacro(msg)
Warning.

26.235.1 Macro Definition Documentation

26.235.1.1 #define GDCM_FUNCTION "<unknown>"

26.235.1.2 #define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)

AssertAlways.

Parameters

<i>arg</i>	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro("my message" && 2 < 3)</code>
------------	---

Referenced by `gdcm::VR::Write()`.

26.235.1.3 #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::GetStream();
        _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()`.

26.235.1.4 #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::GetStream();
        _os << osmacro.str() << "\n\n" << std::endl;
    }
}

```

Debug.

Parameters

<i>msg</i>	message part
------------	--------------

Referenced by `gdcm::ByteValue::ByteValue()`, `gdcm::SequenceOfFragments::Read()`, `gdcm::SequenceOfItems::Read()`, `gdcm::Item::Read()`, `gdcm::VR::Read()`, and `gdcm::ByteValue::SetLength()`.

26.235.1.5 #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::GetStream();
    \
    _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()`, and `gdcM::Item::Read()`.

26.235.1.6 #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::GetStream();
    \
    _os << osmacro.str() << std::endl;
    \
}
}

```

Warning.

Parameters

<i>msg</i>	message part
------------	--------------

Referenced by `gdcM::DataSet::InsertDataElement()`, `gdcM::SequenceOfFragments::Read()`, `gdcM::SequenceOfItems::Read()`, `gdcM::Item::Read()`, `gdcM::Fragment::ReadValue()`, and `gdcM::Item::Write()`.

26.236 gdcMTransferSyntax.h File Reference

Classes

- class `gdcM::TransferSyntax`
Class to manipulate Transfer Syntax.

Namespaces

- namespace `gdcM`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)`

26.237 gdcmTransferSyntaxSub.h File Reference

Classes

- class `gdcm::network::TransferSyntaxSub`

TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

26.238 gdcmType.h File Reference

Classes

- class `gdcm::Type`

Type.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)`

26.239 gdcmTypes.h File Reference

Macros

- `#define UINT32_MAX (4294967295U)`

26.239.1 Macro Definition Documentation

26.239.1.1 `#define UINT32_MAX (4294967295U)`

26.240 gdcmUIDGenerator.h File Reference

Classes

- class `gdcm::UIDGenerator`
Class for generating unique UID.

Namespaces

- namespace `gdcm`

26.241 gdcmUIDs.h File Reference

Classes

- class `gdcm::UIDs`
all known uids

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)`

26.242 gdcmULAction.h File Reference

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

- namespace `gdcm`
- namespace `gdcm::network`

26.243 gdcmULActionAA.h File Reference

Classes

- class `gdcm::network::ULActionAA1`
- class `gdcm::network::ULActionAA2`
- class `gdcm::network::ULActionAA3`
- class `gdcm::network::ULActionAA4`

- class gdcm::network::ULActionAA5
- class gdcm::network::ULActionAA6
- class gdcm::network::ULActionAA7
- class gdcm::network::ULActionAA8

Namespaces

- namespace gdcm
- namespace gdcm::network

26.244 gdcmULActionAE.h File Reference

Classes

- class gdcm::network::ULActionAE1
- class gdcm::network::ULActionAE2
- class gdcm::network::ULActionAE3
- class gdcm::network::ULActionAE4
- class gdcm::network::ULActionAE5
- class gdcm::network::ULActionAE6
- class gdcm::network::ULActionAE7
- class gdcm::network::ULActionAE8

Namespaces

- namespace gdcm
- namespace gdcm::network

26.245 gdcmULActionAR.h File Reference

Classes

- class gdcm::network::ULActionAR1
- class gdcm::network::ULActionAR10
- class gdcm::network::ULActionAR2
- class gdcm::network::ULActionAR3
- class gdcm::network::ULActionAR4
- class gdcm::network::ULActionAR5
- class gdcm::network::ULActionAR6
- class gdcm::network::ULActionAR7
- class gdcm::network::ULActionAR8
- class gdcm::network::ULActionAR9

Namespaces

- namespace gdcm
- namespace gdcm::network

26.246 gdcmULActionDT.h File Reference

Classes

- class gdcm::network::ULActionDT1
- class gdcm::network::ULActionDT2

Namespaces

- namespace gdcm
- namespace gdcm::network

26.247 gdcmULBasicCallback.h File Reference

Classes

- class gdcm::network::ULBasicCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

26.248 gdcmULConnection.h File Reference

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

- namespace gdcm
- namespace gdcm::network

26.249 gdcmULConnectionCallback.h File Reference

Classes

- class gdcm::network::ULConnectionCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

26.250 gdcmULConnectionInfo.h File Reference

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

- namespace gdcm
- namespace gdcm::network

26.251 gdcmULConnectionManager.h File Reference

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

- namespace gdcm
- namespace gdcm::network

26.252 gdcmULEvent.h File Reference

Classes

- class gdcm::network::ULEvent

ULEvent base class for network events.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.253 gdcmULTransitionTable.h File Reference

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

- namespace gdcm
- namespace gdcm::network

26.254 gdcmULWritingCallback.h File Reference

Classes

- class gdcm::network::ULWritingCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

26.255 gdcmUNExplicitDataElement.h File Reference

Classes

- class gdcm::UNExplicitDataElement
- Class to read/write a DataElement as UNExplicit Data Element.*

Namespaces

- namespace gdcm

26.256 gdcmUNExplicitImplicitDataElement.h File Reference

Classes

- class gdcm::UNExplicitImplicitDataElement
- Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:*

Namespaces

- namespace gdcm

26.257 gdcmUnpacker12Bits.h File Reference

Classes

- class gdcm::Unpacker12Bits
Pack/Unpack 12 bits pixel into 16bits.

Namespaces

- namespace gdcm

26.258 gdcmUsage.h File Reference

Classes

- class gdcm::Usage
Usage.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)

26.259 gdcmUserInformation.h File Reference

Classes

- class gdcm::network::UserInformation
UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

Namespaces

- namespace gdcm
- namespace gdcm::network

26.260 gdcmValidate.h File Reference

Classes

- class gdcm::Validate
Validate class.

Namespaces

- namespace gdcm

26.261 gdcmValue.h File Reference

Classes

- class gdcm::Value
Class to represent the value of a Data Element.

Namespaces

- namespace gdcm

26.262 gdcmValueIO.h File Reference

Classes

- class gdcm::ValueIO< TDE, TSwap, TType >
Class to dispatch template calls.

Namespaces

- namespace gdcm

26.263 gdcmVersion.h File Reference

Classes

- class gdcm::Version
major/minor and build version

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)

26.264 gdcmviewer.man File Reference

26.265 gdcmVL.h File Reference

Classes

- class gdcm::VL
Value Length.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)

26.266 gdcmVM.h File Reference

Classes

- class gdcm::VM
Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Namespaces

- namespace gdcm

Macros

- #define TYPETOLENGTH(type, length)

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VM &_val)

26.266.1 Macro Definition Documentation

26.266.1.1 #define TYPETOLENGTH(type, length)

Value:

```
template< > struct VMToLength<VM::type> \
{ enum { Length = length }; };
```

26.267 gdcmVR.h File Reference

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.

Namespaces

- namespace gdcm

Macros

- #define TYPETOENCODING(type, rep, rtype)
- #define VRTypeTemplateCase(type)

Typedefs

- typedef String<'\', 16 > gdcm::AECComp
- typedef String<'\', 64 > gdcm::ASComp
- typedef String<'\', 16 > gdcm::CSComp
- typedef String<'\', 64 > gdcm::DAComp
- typedef String<'\', 64 > gdcm::DTComp
- typedef String<'\', 64 > gdcm::LOComp
- typedef String<'\', 64 > gdcm::LTComp
- typedef String<'\', 64 > gdcm::PNComp
- typedef String<'\', 64 > gdcm::SHComp
- typedef String<'\', 64 > gdcm::STComp
- typedef String<'\', 16 > gdcm::TMComp
- typedef String<'\', 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

Variables

- gdcm::VRBINARY

26.267.1 Macro Definition Documentation

26.267.1.1 #define TYPETOENCODING(type, rep, rtype)

Value:

```
template<> struct VRToEncoding<VR::type> \
{ enum { Mode = VR::rep }; }; \
template<> struct VRToType<VR::type> \
{ typedef rtype Type; };
```

26.267.1.2 #define VRTypeTemplateCase(type)

Value:

```
case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );
```

Referenced by gdcm::VR::GetSize().

26.268 gdcmVR16ExplicitDataElement.h File Reference

Classes

- class gdcm::VR16ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element.

Namespaces

- namespace gdcm

26.269 gdcmWaveform.h File Reference

Classes

- class gdcm::Waveform
Waveform class.

Namespaces

- namespace gdcm

26.270 gdcmWin32.h File Reference

Macros

- #define GDCM_EXPORT

26.270.1 Macro Definition Documentation

26.270.1.1 `#define GDCM_EXPORT`

26.271 `gdcmWriter.h` File Reference

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

- namespace `gdcm`

26.272 `gdcmXMLDictReader.h` File Reference

Classes

- class `gdcm::XMLDictReader`
Class for representing a XMLDictReader.

Namespaces

- namespace `gdcm`

26.273 `gdcmXMLPrivateDictReader.h` File Reference

Classes

- class `gdcm::XMLPrivateDictReader`
Class for representing a XMLPrivateDictReader.

Namespaces

- namespace `gdcm`

26.274 `itkGDCMImageIO2.h` File Reference

Classes

- class `itk::GDCMImageIO2`
ImageIO class for reading and writing DICOM V3.0 and ACR/NEMA (V1.0 & V2.0) images This class is only an adaptor to the gdcm library (currently gdcm 2.0 is used):

Namespaces

- namespace itk

Macros

- #define ITK_GDCM_EXPORT

26.274.1 Macro Definition Documentation

26.274.1.1 #define ITK_GDCM_EXPORT

26.275 README.txt File Reference

26.276 TestsList.txt File Reference

26.277 vtkGDCMImageReader.h File Reference

Classes

- class vtkGDCMImageReader

Namespaces

- namespace gdcmm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

26.277.1 Macro Definition Documentation

26.277.1.1 #define VTK_CMYK 8

26.277.1.2 #define VTK_INVERSE_LUMINANCE 5

26.277.1.3 #define VTK_LOOKUP_TABLE 6

26.277.1.4 #define VTK_YBR 7

26.278 vtkGDCMImageWriter.h File Reference

Classes

- class vtkGDCMImageWriter

26.279 vtkGDCMMedicalImageProperties.h File Reference

Classes

- class vtkGDCMMedicalImageProperties

Namespaces

- namespace gdcm

26.280 vtkGDCMPolyDataReader.h File Reference

Classes

- class vtkGDCMPolyDataReader

Namespaces

- namespace gdcm

26.281 vtkGDCMPolyDataWriter.h File Reference

Classes

- class vtkGDCMPolyDataWriter

Namespaces

- namespace gdcm

26.282 vtkGDCMTesting.h File Reference

Classes

- class vtkGDCMTesting

26.283 vtkGDCMThreadedImageReader.h File Reference

Classes

- class vtkGDCMThreadedImageReader

26.284 vtkGDCMThreadedImageReader2.h File Reference

Classes

- class vtkGDCMThreadedImageReader2

26.285 vtkImageColorViewer.h File Reference

Classes

- class vtkImageColorViewer

26.286 vtkImageMapToColors16.h File Reference

Classes

- class vtkImageMapToColors16

26.287 vtkImageMapToWindowLevelColors2.h File Reference

Classes

- class vtkImageMapToWindowLevelColors2

26.288 vtkImagePlanarComponentsToComponents.h File Reference

Classes

- class vtkImagePlanarComponentsToComponents

26.289 vtkImageRGBToYBR.h File Reference

Classes

- class vtkImageRGBToYBR

26.290 `vtkImageYBRToRGB.h` File Reference

Classes

- class `vtkImageYBRToRGB`

26.291 `vtkLookupTable16.h` File Reference

Classes

- class `vtkLookupTable16`

26.292 `vtkRTStructSetProperties.h` File Reference

Classes

- class `vtkRTStructSetProperties`

Chapter 27

Example Documentation

27.1 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);
    }
}

```

27.2 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)
    {
        gdcmm.Global global = gdcmm.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 = gdcmm.Filename.Join(gdcmm.Testing.GetSourceDirectory(), "
            /Testing/Source/Data/certificate.pem" );
        gdcmm.CryptographicMessageSyntax cms = new gdcmm.CryptographicMessageSyntax(
            ;
        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;
    }
}

```

27.3 CastConvertPhilips.py

```
#####
#
# Program: GDCM (Grassroots DICOM). A DICOM library
#
# Copyright (c) 2006-2011 Mathieu Malaterre
# All rights reserved.
# See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
# This software is distributed 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:

python --public /path/to/directory/
or
python --private /path/to/directory/

python --public --extension bak /path/to/directory/

rename -f 's/\.bak$/' *.bak

TODO:
http://docs.python.org/library/optparse.html#module-optparse
"""

import vtkgdcml
import vtk
import sys
import gdcm

def ProcessOneFilePublic(filename, outfile, tmpfile):
    gdcml.ImageHelper.SetForceRescaleInterceptSlope(
        True)
    vtkreader = vtkgdcml.vtkGDCMLImageReader()
    vtkreader.SetFileName( filename )
    vtkreader.Update()

    cast = vtk.vtkImageCast()
    cast.SetInput( vtkreader.GetOutput() )
    cast.SetOutputScalarTypeToUnsignedShort()

    # vtkGDCMLImageWriter does not support Sequence, so let's write a tmp file
    # first:
    # Some operation will actually be discarded (we simply need a temp storage)
    vtkwriter = vtkgdcml.vtkGDCMLImageWriter()
    vtkwriter.SetFileName( tmpfile )
    vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
    vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
    print "Format:", vtkreader.GetImageFormat()
    vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
    vtkwriter.SetInput( cast.GetOutput() )
    #vtkwriter.Update()
    vtkwriter.Write()

    # ok now rewrite the exact same file as the original (keep all info)
    # but use the Pixel Data Element from the written file
    tmpreader = gdcm.ImageReader()
    tmpreader.SetFileName( tmpfile )
    if not tmpreader.Read():
        sys.exit(1)

    reader = gdcm.Reader()
    reader.SetFileName( filename )
    if not reader.Read():
        sys.exit(1)

    # Make sure to remove Slope/Rescale to avoid re-execution
    ds = reader.GetFile().GetDataSet()
    tags = [
        gdcm.Tag(0x0028, 0x1052),
        gdcm.Tag(0x0028, 0x1053),
        gdcm.Tag(0x0028, 0x1053),
    ]
    for tag in tags:
```

```

    ds.Remove( tag )

writer = gdcm.ImageWriter()
writer.SetFileName( outfilename )
# Pass image from vtk written file
writer.SetImage( tmpreader.GetImage() )
# pass dataset from initial 'reader'
writer.SetFile( reader.GetFile() )
if not writer.Write():
    sys.exit(1)

def ProcessOneFilePrivate(filename, outfilename, tmpfile):
    vtkreader = vtkgdcm.vtkGDCMImageReader()
    vtkreader.SetFileName( filename )
    vtkreader.Update()

    # (2005,1409)    DS      4      0.0
    # (2005,140a)    DS     16     1.52283272283272

    # (2005,0014)    LO     26     Philips MR Imaging DD 005
    tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005"
    )
    tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005"
    )

    # Need to access some private tags, reread the file (for now):
    reader = gdcm.Reader()
    reader.SetFileName( filename )
    if not reader.Read():
        sys.exit(1)

    ds = reader.GetFile().GetDataSet()

    el1 = ds.GetDataElement( tag1 )
    el2 = ds.GetDataElement( tag2 )

    #pf = gdcm.PythonFilter()
    #pf.SetFile( reader.GetFile() )
    #print el1.GetTag()

    print el1.GetByteValue()
    v1 = eval(el1.GetByteValue().GetBuffer())
    print el2.GetByteValue()
    v2 = eval(el2.GetByteValue().GetBuffer())

    print v1
    shift = v1
    print v2
    scale = v2

    ss = vtk.vtkImageShiftScale()
    ss.SetInput( vtkreader.GetOutput() )
    # because VTK image shift / scale convention is inverted from DICOM make sure
    # shift is 0
    assert shift == 0
    ss.SetShift( shift )
    ss.SetScale( scale )
    ss.SetOutputScalarTypeToUnsignedShort ()
    ss.Update()

    # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file
    # first:
    # Some operation will actually be discarded (we simply need a temp storage)
    vtkwriter = vtkgdcm.vtkGDCMImageWriter()
    vtkwriter.SetFileName( tmpfile )
    vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
    vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
    vtkwriter.SetImageFormat( reader.GetImageFormat() )
    # do not pass shift/scale again
    vtkwriter.SetInput( ss.GetOutput() )
    #vtkwriter.Update()
    vtkwriter.Write()

    # ok now rewrite the exact same file as the original (keep all info)
    # but use the Pixel Data Element from the written file
    tmpreader = gdcm.ImageReader()
    tmpreader.SetFileName( tmpfile )

```

```

if not tmpreader.Read():
    sys.exit(1)

writer = gdcm.ImageWriter()
writer.SetFileName( outfilename )
# Pass image from vtk written file
writer.SetImage( tmpreader.GetImage() )
# pass dataset from initial 'reader'
writer.SetFile( reader.GetFile() )
if not writer.Write():
    sys.exit(1)

if __name__ == "__main__":

    gdcm.Trace.DebugOff()
    gdcm.Trace.WarningOff()
    #filename = sys.argv[1]
    #outfilename = sys.argv[2]
    tmpfile = "/tmp/philips_rescaled.dcm"
    #ProcessOneFile( filename, outfilename, tmpfile )
    rescaletype = sys.argv[1]
    assert rescaletype == "--public" or rescaletype == "--private"
    dirname = sys.argv[2]
    d = gdcm.Directory()
    d.Load( dirname )

    for f in d.GetFileNames():
        #print f
        ProcessOneFilePublic( f, f + ".bak", tmpfile )

print "success"

```

27.4 ChangeSequenceUltrasound.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmSmartPointer.h"
#include "gdcmDataSetHelper.h"

/*
./ChangeSequenceUltrasound gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

This is the exact C++ translation of the original python example:
ManipulateSequence.py
*/

int main(int argc, char* argv[] )
{
    if( argc < 0 )
    {
        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();

```

```

gdcmm::DataSet &ds = file.GetDataSet();
gdcmm::Tag tsis(0x0008,0x2112); // SourceImageSequence
if ( ds.FindDataElement( tsis ) )
{
    const gdcmm::DataElement &sis = ds.GetDataElement
        ( tsis );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems>
        sqsis = sis.GetValueAsSQ();
    if ( sqsis && sqsis->GetNumberOfItems() )
    {
        gdcmm::Item &item1 = sqsis->GetItem(1);
        gdcmm::DataSet &nestedds = item1.GetNestedDataSet
            ();
        gdcmm::Tag tprcs(0x0040,0xa170); //
            PurposeOfReferenceCodeSequence
        if( nestedds.FindDataElement( tprcs ) )
        {
            const gdcmm::DataElement &prcs = nestedds.
                GetDataElement( tprcs );
            gdcmm::SmartPointer<gdcmm::SequenceOfItems>
                sqprcs = prcs.GetValueAsSQ();
            if ( sqprcs && sqprcs->GetNumberOfItems() )
            {
                gdcmm::Item &item2 = sqprcs->GetItem(1);
                gdcmm::DataSet &nestedds2 = item2.GetNestedDataSet
                    ();
                // (0008,0104) LO [Uncompressed predecessor] # 24, 1
                CodeMeaning
                gdcmm::Tag tcm(0x0008,0x0104);
                if( nestedds2.FindDataElement( tcm ) )
                {
                    gdcmm::DataElement cm = nestedds2.GetDataElement
                        ( tcm );
                    std::string mystr = "GDCM was here";
                    cm.SetByteValue( mystr.c_str(), mystr.size() );
                    nestedds2.Replace( cm );
                }
            }
        }
    }
}

gdcmm::Writer writer;
writer.SetFile( file );
writer.SetFileName( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

27.5 CheckBigEndianBug.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * WARNING: This is a dev tool, do not use !
 *
 * Usage: after a gdcmmconv, you would like to know if the conversion process is
        acceptable
 * sometime a vbindiff is acceptable, sometime it is not. In the case of the
        famous Philips
 * Little/Big Endian Explicit Transfer Syntax it is not easy to compare two
        files. However

```

```

* this only impact byte ordering, thus we can compute byte-independant
  information to still
* compare the files.
*/

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmSystem.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input1.dcm input2.dcm" << std::endl;
        return 1;
    }
    const char *filename1 = argv[1];
    const char *filename2 = argv[2];

    gdcm::ImageReader reader1;
    reader1.SetFileName( filename1 );
    if( !reader1.Read() )
    {
        std::cerr << "Could not read: " << filename1 << std::endl;
        return 1;
    }

    gdcm::ImageReader reader2;
    reader2.SetFileName( filename2 );
    if( !reader2.Read() )
    {
        std::cerr << "Could not read: " << filename2 << std::endl;
        return 1;
    }

    // TODO: need a DataSet== operator implementation

    std::cout << "Both files can be read and looks like DICOM" << std::endl;

    size_t s1 = gdcm::System::FileSize(filename1);
    size_t s2 = gdcm::System::FileSize(filename2);

    if( s1 != s2 )
    {
        std::cout << "Size mismatch: " << s1 << " != " << s2 << std::endl;
        return 1;
    }
    else
    {
        std::cout << "Size match: " << s1 << " = " << s2 << std::endl;
    }

    std::ifstream is1( filename1 );
    char *buffer1 = new char[s1];
    is1.read(buffer1, s1);

    std::ifstream is2( filename2 );
    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;
}

```

27.6 ClinicalTrialAnnotate.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.
=====*/

/*
 * Dummy implementation of C.7.1.3 Clinical Trial Subject Module
 *
 * Usage:
 * ClinicalTrialAnnotate gdcmlData/012345.002.050.dcm out.dcm
 */

#include "gdcmlReader.h"
#include "gdcmlWriter.h"
#include "gdcmlAnonymizer.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];

    gdcml::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcml::Reader is a gdcml::File
    //gdcml::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    //gdcml::DataSet &ds = file.GetDataSet();

    gdcml::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( gdcml::Tag(0x12,0x10), "BigCompany name" );
}

```

```

// <entry group="0012" element="0020" vr="LO" vm="1" name="Clinical Trial
  Protocol ID"/>
ano.Replace( gdcmm::Tag(0x12,0x20), "My Clinical Trial
  Protocol ID" );
// <entry group="0012" element="0021" vr="LO" vm="1" name="Clinical Trial
  Protocol Name"/>
ano.Replace( gdcmm::Tag(0x12,0x21), "My Clinical Trial
  Protocol Name" );
// <entry group="0012" element="0030" vr="LO" vm="1" name="Clinical Trial
  Site ID"/>
ano.Replace( gdcmm::Tag(0x12,0x30), "My Clinical Trial Site ID
  " );
// <entry group="0012" element="0031" vr="LO" vm="1" name="Clinical Trial
  Site Name"/>
ano.Replace( gdcmm::Tag(0x12,0x31), "My Clinical Trial Site
  Name" );
// <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial
  Subject ID"/>
ano.Replace( gdcmm::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( gdcmm::Tag(0x12,0x42), "My Clinical Trial Subject
  Reading ID" );

gdcmm::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

27.7 ClinicalTrialIdentificationWorkflow.cs

This is a C# example on how to use gdcmm::Anonymizer

```

/*=====
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.

=====*/
/*
* Typical usage on UNIX:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcmm/debug-gcc/bin
* $ mono bin/ClinicalTrialIdentificationWorkflow.exe input_dir output_dir
*/
using System;
using gdcmm;

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.FileMetaInformation fmi = ano.GetFile().GetHeader();
// The following three lines make sure to regenerate any value:
fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

Writer writer = new Writer();
writer.SetFileName( outfilename );
writer.SetFile( ano.GetFile() );
ret = writer.Write();
if( !ret )
{
    return false;
}

return true;
}

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My ClinicalTrial
        App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot()
        );

    gdcm.Global global = gdcm.Global.GetInstance();
    if( !global.LoadResourcesFiles() )
    {
        System.Console.WriteLine( "Could not LoadResourcesFiles" );
        return 1;
    }

    if( args.Length != 2 )
    {
        System.Console.WriteLine( "Usage:" );
        System.Console.WriteLine( "ClinicalTrialIdentificationWorkflow input_dir
            output_dir" );
        return 1;
    }
    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist.
            Sorry" );
        return 1;
    }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
    {
        System.Console.WriteLine( "Output directory: " + dir2 + " does not exist.
            Sorry" );
        return 1;
    }
}

```

```

// Recursively search all file within this toplevel directory:
Directory d = new Directory();
uint nfiles = d.Load( dir1, true );
if(nfiles == 0) return 1;

// Let's use the pre-shipped certificate of GDCM.
string certpath = gdcml.Filename.Join(gdcml.Testing.GetSourceDirectory(), "
    /Testing/Source/Data/certificate.pem" );
gdcml.CryptographicMessageSyntax cms = new gdcml.CryptographicMessageSyntax()
;
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);

// Explicitely specify the Cryptographic Message Syntax to use:
ano.SetCryptographicMessageSyntax( cms );

// 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( ano , filename, outfilename ) )
    {
        System.Console.WriteLine( "Could not process filename: " + filename );
        return 1;
    }
}

return 0;
}
}

```

27.8 CompressImage.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 "gdcmlImageReader.h"
#include "gdcmlImage.h"
#include "gdcmlWriter.h"
#include "gdcmlAttribute.h"
#include "gdcmlImageWriter.h"
#include "gdcmlImageChangeTransferSyntax.h"

#include <iostream>
#include <fstream>

```

```

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    //gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    //gdcm::DataSet &ds = file.GetDataSet();

    const gdcm::Image &image = reader.GetImage();
    image.Print( std::cout );

    gdcm::ImageChangeTransferSyntax change;
    change.SetTransferSyntax(
        gdcm::TransferSyntax::JPEG2000Lossless );
    change.SetTransferSyntax(
        gdcm::TransferSyntax::JPEGLosslessProcess14_1
    );
    //change.SetTransferSyntax( gdcm::TransferSyntax::JPEGBaselineProcess1 );
    //change.SetTransferSyntax( image.GetTransferSyntax() );
    change.SetInput( image );
    bool b = change.Change();
    if( !b )
    {
        std::cerr << "Could not change the Transfer Syntax" << std::endl;
        return 1;
    }

    //std::ofstream out( outfile );
    //image.GetBuffer2(out);
    //out.close();
    gdcm::ImageWriter writer;
    writer.SetImage( change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfile );
    if( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```

27.9 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;
    }
}

```

27.10 Convert16BitsTo8Bits.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"

#include "gdcmTesting.h"
// The following file is 16/16/15 but the scalar range of the image is [0,192]
// it could be safely stored as 8bits instead:
// gdcmData/012345.002.050.dcm

int main(int, char *[])
{
    const char *directory = gdcm::Testing::GetDataRoot(
    );
    if(!directory) return 1;
    std::string file = std::string(directory) + "/012345.002.050.dcm";
    std::cout << file << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New
    ();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageCast *cast = vtkImageCast::New();
    cast->SetInput( reader->GetOutput() );
    cast->SetOutputScalarTypeToUnsignedChar();

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
    ();
    writer->SetFileName( "/tmp/cast.dcm" );
    writer->SetInput( cast->GetOutput() );
    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;
}

```

27.11 ConvertMPL.py

```

#####
#
# Program: GDCM (Grassroots DICOM). A DICOM library
#
# Copyright (c) 2006-2011 Mathieu Malaterre
# All rights reserved.
# See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
# This software is distributed WITHOUT ANY WARRANTY; without even
# the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

```

#     PURPOSE.  See the above copyright notice for more information.
#
#####

"""
display a DICOM image with matplotlib via numpy

Caveats:
- Does not support UINT12/INT12

Usage:

python ConvertNumpy.py "IM000000"

Thanks:
plotting example - Ray Schumacher 2009
"""

import gdcm
import numpy
from pylab import *

def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
                 gdcm.PixelFormat.INT8 :numpy.uint8,
                 gdcm.PixelFormat.UINT16 :numpy.uint16,
                 gdcm.PixelFormat.INT16 :numpy.int16,
                 gdcm.PixelFormat.UINT32 :numpy.uint32,
                 gdcm.PixelFormat.INT32 :numpy.int32,
                 gdcm.PixelFormat.FLOAT32 :numpy.float32,
                 gdcm.PixelFormat.FLOAT64 :numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat().GetScalarType()
    print 'pf', pf
    print image.GetPixelFormat().GetScalarTypeAsString()
    assert pf in get_gdcm_to_numpy_typemap().keys(), \
        "Unsupported array type %s"%pf
    d = image.GetDimension(0), image.GetDimension(1)
    print 'Image Size: %d x %d' % (d[0], d[1])
    dtype = get_numpy_array_type(pf)
    gdcm_array = image.GetBuffer()
    ## use float for accurate scaling
    result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
    ## optional gamma scaling
    #maxV = float(result[result.argmax()])
    #result = result + .5*(maxV-result)
    #result = numpy.log(result+50) ## apprx background level
    result.shape = d
    return result

if __name__ == "__main__":
    import sys
    r = gdcm.ImageReader()
    filename = sys.argv[1]
    r.SetFileName( filename )
    if not r.Read(): sys.exit(1)
    numpy_array = gdcm_to_numpy( r.GetImage() )

    subplot(111)# one plot, on left
    title(filename)
    ## many colormaps are available
    imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
    ## set the plot sizes and placement
    subplots_adjust(bottom=0.1, right=0.8, top=0.9)
    cax = axes([0.85, 0.1, 0.075, 0.8])
    colorbar(cax=cax)
    title('values')
    get_current_fig_manager().window.title('plot')
    show()

```

27.12 ConvertMultiFrameToSingleFrame.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkStringArray.h"

#include "gdcmTesting.h"
#include "gdcmFilenameGenerator.h"

int main(int argc, char *argv[])
{
    std::string filename;
    if( argc <= 1 )
    {
        const char *directory = gdcm::Testing::GetDataRoot
            ();
        if(!directory) return 1;
        std::string file = std::string(directory) + "/US-PAL-8-10x-echo.dcm";
        filename = file;
    }
    else
    {
        filename = argv[1];
    }
    std::cout << "file: " << filename << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New
        ();
    reader->SetFileName( filename.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    int dims[3];
    reader->GetOutput()->GetDimensions( dims );

    std::ostringstream os;
    os << "singleframe";
    os << "%04d.dcm";
    gdcm::FilenameGenerator fg;
    fg.SetPattern( os.str().c_str() );
    unsigned int nfiles = dims[2];
    fg.SetNumberOfFileNames( nfiles );
    bool b = fg.Generate();
    if( !b )
    {
        std::cerr << "FilenameGenerator::Generate() failed" << std::endl;
        return 1;
    }
    if( !fg.GetNumberOfFileNames() )
    {
        std::cerr << "FilenameGenerator::Generate() failed somehow..." << std::endl;
        return 1;
    }

    // By default write them as Secondary Capture (for portability)
    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
        ();
    vtkStringArray *filenames = vtkStringArray::New();
    for(unsigned int i = 0; i < fg.GetNumberOfFileNames(); ++
        i)
    {
        filenames->InsertNextValue( fg.GetFilename(i) );
    }
    assert( filenames->GetNumberOfValues() == (int)fg.GetNumberOfFileNames
        () );
}

```

```

writer->SetFileNames( filenames );
filenames->Delete();
writer->SetFileDimensionality( 2 );
writer->SetInput( reader->GetOutput() );
writer->SetImageFormat( reader->GetImageFormat() );
writer->Write();

reader->Delete();
writer->Delete();

return 0;
}

```

27.13 ConvertNumpy.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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 module add support for converting a gdcm.Image to a numpy array.

Caveats:
- Does not support UINT12/INT12

Removed:
- float16 is defined in GDCM API but no implementation exist for it ...
"""

import gdcm
import numpy

def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
                 gdcm.PixelFormat.INT8 :numpy.uint8,
                 #gdcm.PixelFormat.UINT12 :numpy.uint12,
                 #gdcm.PixelFormat.INT12 :numpy.int12,
                 gdcm.PixelFormat.UINT16 :numpy.uint16,
                 gdcm.PixelFormat.INT16 :numpy.int16,
                 gdcm.PixelFormat.UINT32 :numpy.uint32,
                 gdcm.PixelFormat.INT32 :numpy.int32,
                 #gdcm.PixelFormat.FLOAT16:numpy.float16,
                 gdcm.PixelFormat.FLOAT32:numpy.float32,
                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat()

    assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \
        "Unsupported array type %s"%pf

    shape = image.GetDimension(0) * image.GetDimension(1),
    pf.GetSamplesPerPixel()
    if image.GetNumberOfDimensions() == 3:
        shape = shape[0] * image.GetDimension(2), shape[1]

    dtype = get_numpy_array_type(pf.GetScalarType())
    gdcm_array = image.GetBuffer()
    result = numpy.frombuffer(gdcm_array, dtype=dtype)

```

```

        result.shape = shape
        return result

if __name__ == "__main__":
    import sys
    r = gdcm.ImageReader()
    filename = sys.argv[1]
    r.SetFileName( filename )
    if not r.Read():
        sys.exit(1)

    numpy_array = gdcm_to_numpy( r.GetImage() )
    print numpy_array

```

27.14 ConvertPIL.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed WITHOUT ANY WARRANTY; without even
#   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#   PURPOSE. See the above copyright notice for more information.
#
#####

"""
save a DICOM image with PIL via numpy

Caveats:
- Does not support UINT12/INT12

Usage:

python ConvertNumpy.py "IM000000"

Thanks:
    plotting example - Ray Schumacher 2009
"""

import gdcm
import numpy
from PIL import Image, ImageOps

def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
                 gdcm.PixelFormat.INT8 :numpy.uint8,
                 gdcm.PixelFormat.UINT16 :numpy.uint16,
                 gdcm.PixelFormat.INT16 :numpy.int16,
                 gdcm.PixelFormat.UINT32 :numpy.uint32,
                 gdcm.PixelFormat.INT32 :numpy.int32,
                 gdcm.PixelFormat.FLOAT32:numpy.float32,
                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat().GetScalarType()
    print 'pf', pf
    print image.GetPixelFormat().GetScalarTypeAsString()
    assert pf in get_gdcm_to_numpy_typemap().keys(), \
        "Unsupported array type %s"%pf
    d = image.GetDimension(0), image.GetDimension(1)
    print 'Image Size: %d x %d' % (d[0], d[1])
    dtype = get_numpy_array_type(pf)
    gdcm_array = image.GetBuffer()

```

```

    result = numpy.frombuffer(gdcm_array, dtype=dtype)
    maxV = float(result[result.argmax()])
    ## linear gamma adjust
    #result = result + .5*(maxV-result)
    ## log gamma
    result = numpy.log(result+50) ## 50 is apprxx background level
    maxV = float(result[result.argmax()])
    result = result*(2.**8/maxV) ## histogram stretch
    result.shape = d
    return result

if __name__ == "__main__":
    import sys
    r = gdcm.ImageReader()
    filename = sys.argv[1]
    r.SetFileName( filename )
    if not r.Read(): sys.exit(1)
    numpy_array = gdcm_to_numpy( r.GetImage() )
    ## L is 8 bit grey
    ## http://www.pythonware.com/library/pil/handbook/concepts.htm
    pilImage = Image.frombuffer('L',
                                numpy_array.shape,
                                numpy_array.astype(numpy.uint8),
                                'raw','L',0,1)

    ## cutoff removes background noise and spikes
    pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
    pilImage.save(sys.argv[1]+' .jpg')

```

27.15 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();
    luminance->SetInput( reader->GetOutput() );

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New(
    );
    writer->SetFileName( "/tmp/bla.dcm" );
    writer->SetInput( luminance->GetOutput() );
    //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;
}

```

27.16 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, barray->GetValue(i) );
    }

    vtkImageData *copy = vtkImageData::New();
    copy->SetScalarType( VTK_UNSIGNED_CHAR );
    copy->SetExtent( reader->GetOutput()->GetExtent() );
    copy->AllocateScalars();

    //uarray->Print( std::cout );
    //copy->GetPointData()->GetScalars()->Print( std::cout );
    copy->GetPointData()->SetScalars( uarray );
    uarray->Delete();
}

```

```

vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
();
writer->SetFileName( outfilename );
//writer->SetInput( cast->GetOutput() );
writer->SetInput( copy );
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;
}

```

27.17 ConvertToQImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * This example shows how to setup the pipeline from a gdcm::ImageReader into a
 * Qt QImage data structure.
 * It only handles 2D image.
 *
 * Ref:
 * http://doc.trolltech.com/4.5/qimage.html
 *
 * Usage:
 * ConvertToQImage gdcmData/012345.002.050.dcm output.png
 *
 * Thanks:
 * Sylvain ADAM (sylvain51 hotmail com) for contributing this example
 */

#include "gdcmImageReader.h"
#include <QImage>
#include <QImageWriter>

bool ConvertToFormat_RGB888(gdcm::Image const & gimage, char *buffer
, QImage* &imageQt)
{
    const unsigned int* dimension = gimage.GetDimensions();

    unsigned int dimX = dimension[0];
    unsigned int dimY = dimension[1];

    gimage.GetBuffer(buffer);

    // Let's start with the easy case:
    if( gimage.GetPhotometricInterpretation() ==
        gdcm::PhotometricInterpretation::RGB )
    {
        if( gimage.GetPixelFormat() != gdcm::PixelFormat::UINT8
        )
        {
            return false;
        }
        unsigned char *ubuffer = (unsigned char*)buffer;
        // QImage::Format_RGB888 13 The image is stored using a 24-bit RGB format
    }
}

```

```

        (8-8-8).
        imageQt = new QImage((unsigned char *)ubuffer, dimX, dimY, 3*dimX,
                             QImage::Format_RGB888);
    }
    else if( gimage.GetPhotometricInterpretation() ==
             gdcm::PhotometricInterpretation::MONOCHROME2
            )
    {
        if( gimage.GetPixelFormat() == gdcm::PixelFormat::UINT8
            )
        {
            // We need to copy each individual 8bits into R / G and B:
            unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
            unsigned char *pubuffer = ubuffer;
            for(unsigned int i = 0; i < dimX*dimY; i++)
            {
                *pubuffer++ = *buffer;
                *pubuffer++ = *buffer;
                *pubuffer++ = *buffer++;
            }

            imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
        }
        else if( gimage.GetPixelFormat() == gdcm::PixelFormat::INT16
                )
        {
            // We need to copy each individual 16bits into R / G and B (truncate
            // value)
            short *buffer16 = (short*)buffer;
            unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
            unsigned char *pubuffer = ubuffer;
            for(unsigned int i = 0; i < dimX*dimY; i++)
            {
                // Scalar Range of gdcmData/012345.002.050.dcm is [0,192], we could
                // simply do:
                // *pubuffer++ = *buffer16;
                // *pubuffer++ = *buffer16;
                // *pubuffer++ = *buffer16;
                // instead do it right:
                *pubuffer++ = std::min(255, (32768 + *buffer16) / 255);
                *pubuffer++ = std::min(255, (32768 + *buffer16) / 255);
                *pubuffer++ = std::min(255, (32768 + *buffer16) / 255);
                buffer16++;
            }

            imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
        }
        else
        {
            std::cerr << "Pixel Format is: " << gimage.GetPixelFormat()
                      << std::endl;
            return false;
        }
    }
    else
    {
        std::cerr << "Unhandled PhotometricInterpretation: " << gimage.
            GetPhotometricInterpretation() << std::endl;
        return false;
    }

    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader ir;
    ir.SetFileName( filename );
    if(!ir.Read())
    {
        //Read failed
        return 1;
    }
}

```

```

std::cout<<"Getting image from ImageReader..."<<std::endl;

const gdcm::Image &gimage = ir.GetImage();
//This buffer has been declared elsewhere
char *buffer = new char[gimage.GetBufferLength()];

QImage *imageQt = NULL;
if( !ConvertToFormat_RGB888( gimage, buffer, imageQt ) )
{
    return 1;
}

QImageWriter writer;
writer.setFormat("png");
writer.setFileName( outfilename );
if( !writer.write( *imageQt ) )
{
    return 1;
}

// delete[] buffer;

return 0;
}

```

27.18 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.rgba 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);

    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(0x7fe0,0
    x0010) );
pixeldata.SetByteValue( buf, len );
image.SetDataElement( pixeldata );

writer.SetFileName( outfile );
if( !writer.Write() )
{
    return 1;
}
delete[] buf;

return 0;
}

```

27.19 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 *outfile = argv[2];

    size_t len = gdcm::System::FileSize(filename);
    std::ifstream is(filename);

    char * buf = new char[len];
    is.read(buf, len);

    gdcm::ImageWriter writer;

```

```

gdcmm::Image &image = writer.GetImage();
image.SetNumberOfDimensions( 2 );
unsigned int dims[3] = {};
dims[0] = 380;
dims[1] = 287;
image.SetDimensions( dims );
gdcmm::PixelFormat pf = gdcmm::PixelFormat::UINT8
;
pf.SetSamplesPerPixel( 4 );
image.SetPixelFormat( pf );
gdcmm::PhotometricInterpretation pi =
    gdcmm::PhotometricInterpretation::CMYK;
image.SetPhotometricInterpretation( pi );
image.SetTransferSyntax(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian
);

gdcmm::DataElement pixeldata( gdcmm::Tag(0x7fe0,0
x0010) );
pixeldata.SetByteValue( buf, len );
image.SetDataElement( pixeldata );

writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}
delete[] buf;

return 0;
}

```

27.20 CreateJPIPDataSet.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example was created during the GSOC 2011 project for
 * JPIP
 */
#include "gdcmmAnonymizer.h"
#include "gdcmmWriter.h"
#include "gdcmmUIDGenerator.h"
#include "gdcmmFile.h"
#include "gdcmmTag.h"
#include "gdcmmSystem.h"
#include "gdcmmAttribute.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];

    gdcmm::Writer w;
    gdcmm::File &file = w.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    //w.SetCheckFileMetaInformation( true );
    w.SetFileName( outfilename );

    file.GetHeader().SetDataSetTransferSyntax(
        gdcmm::TransferSyntax::JPIPReferenced );

```

```

gdcmm::Anonymizer anon;
anon.SetFile( file );

gdcmm::MediaStorage ms =
    gdcmm::MediaStorage::SecondaryCaptureImageStorage
    ;

gdcmm::UIDGenerator gen;
anon.Replace( gdcmm::Tag(0x0008,0x16), ms.GetString()
    );
std::cout << ms.GetString() << std::endl;
anon.Replace( gdcmm::Tag(0x0008,0x18), gen.Generate()
    );
//
anon.Replace( gdcmm::Tag(0x0010,0x10), "JPIP^EXAMPLE" );
anon.Replace( gdcmm::Tag(0x0010,0x20), "012345" );
anon.Empty( gdcmm::Tag(0x0010,0x30) );
anon.Empty( gdcmm::Tag(0x0010,0x40) );
anon.Empty( gdcmm::Tag(0x0008,0x20) );
anon.Empty( gdcmm::Tag(0x0008,0x30) );
anon.Empty( gdcmm::Tag(0x0008,0x90) );
anon.Empty( gdcmm::Tag(0x0020,0x10) );
anon.Empty( gdcmm::Tag(0x0020,0x11) );
anon.Empty( gdcmm::Tag(0x0008,0x50) );
anon.Empty( gdcmm::Tag(0x0020,0x0013) );
anon.Replace( gdcmm::Tag(0x0020,0xd), gen.Generate() )
    ;
anon.Replace( gdcmm::Tag(0x0020,0xe), gen.Generate() )
    ;
anon.Replace( gdcmm::Tag(0x0008,0x64), "WSD " );

gdcmm::Attribute<0x0028,0x7FE0> at;
at.SetValue( "http://dicom.example.com/jpipserver.cgi?target=img.jp2"
    );
ds.Insert( at.GetAsDataElement() );

// Need to retrieve the PixelFormat information from the given file
if (!w.Write() )
{
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

27.21 CreateRAWStorage.py

```

#####
#
#   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.
#
#####

"""
    <uid value="1.2.840.10008.5.1.4.1.1.66" name="Raw Data Storage" type="SOP
    Class" part="PS 3.4" retired="false"/>
"""

import gdcmm
import sys,os

if __name__ == "__main__":
    r = gdcmm.Reader()
    # Will require Testing...
    dataroot = gdcmm.Testing.GetDataRoot()
    filename = os.path.join( dataroot, '012345.002.050.dcm' )

```

```

r.SetFileName( filename )
r.Read()
f = r.GetFile()
ds = f.GetDataSet()

uid = "1.2.840.10008.5.1.4.1.1.66"
# f = gdcm.File()
# ds = f.GetDataSet()
de = gdcm.DataElement( gdcm.Tag(0x0008,0x0016) )
de.SetByteValue( uid, gdcm.VL(len(uid)) )
vr = gdcm.VR( gdcm.VR.UI )
de.SetVR( vr )
ds.Replace( de )

ano = gdcm.Anonymizer()
ano.SetFile( r.GetFile() )
ano.RemovePrivateTags()
ano.RemoveGroupLength()
taglist = [
gdcm.Tag(0x0008,0x0008),
gdcm.Tag(0x0008,0x0022),
gdcm.Tag(0x0008,0x0032),
gdcm.Tag(0x0008,0x2111),
gdcm.Tag(0x0008,0x1150),
gdcm.Tag(0x0008,0x1155),
gdcm.Tag(0x0008,0x0100),
gdcm.Tag(0x0008,0x0102),
gdcm.Tag(0x0008,0x0104),
gdcm.Tag(0x0040,0xa170),
gdcm.Tag(0x0008,0x2112),
gdcm.Tag(0x0008,0x0100),
gdcm.Tag(0x0008,0x0102),
gdcm.Tag(0x0008,0x0104),
gdcm.Tag(0x0008,0x9215),
gdcm.Tag(0x0018,0x0010),
gdcm.Tag(0x0018,0x0022),
gdcm.Tag(0x0018,0x0050),
gdcm.Tag(0x0018,0x0060),
gdcm.Tag(0x0018,0x0088),
gdcm.Tag(0x0018,0x0090),
gdcm.Tag(0x0018,0x1040),
gdcm.Tag(0x0018,0x1100),
gdcm.Tag(0x0018,0x1110),
gdcm.Tag(0x0018,0x1111),
gdcm.Tag(0x0018,0x1120),
gdcm.Tag(0x0018,0x1130),
gdcm.Tag(0x0018,0x1150),
gdcm.Tag(0x0018,0x1151),
gdcm.Tag(0x0018,0x1152),
gdcm.Tag(0x0018,0x1160),
gdcm.Tag(0x0018,0x1190),
gdcm.Tag(0x0018,0x1210),
gdcm.Tag(0x0020,0x0012),
gdcm.Tag(0x0020,0x0032),
gdcm.Tag(0x0020,0x0037),
gdcm.Tag(0x0020,0x1041),
gdcm.Tag(0x0020,0x4000),
gdcm.Tag(0x0028,0x0002),
gdcm.Tag(0x0028,0x0004),
gdcm.Tag(0x0028,0x0010),
gdcm.Tag(0x0028,0x0011),
gdcm.Tag(0x0028,0x0030),
gdcm.Tag(0x0028,0x0100),
gdcm.Tag(0x0028,0x0101),
gdcm.Tag(0x0028,0x0102),
gdcm.Tag(0x0028,0x0103),
gdcm.Tag(0x0028,0x1052),
gdcm.Tag(0x0028,0x1053),
gdcm.Tag(0x0028,0x2110),
gdcm.Tag(0x0028,0x2112),
gdcm.Tag(0x7fe0,0x0010),
gdcm.Tag(0x0018,0x0020),
gdcm.Tag(0x0018,0x0021),
gdcm.Tag(0x0018,0x0023),
gdcm.Tag(0x0018,0x0025),
gdcm.Tag(0x0018,0x0080),
gdcm.Tag(0x0018,0x0081),
gdcm.Tag(0x0018,0x0083),
gdcm.Tag(0x0018,0x0084),
gdcm.Tag(0x0018,0x0085),
gdcm.Tag(0x0018,0x0086),

```

```

gdcM.Tag(0x0018,0x0087),
gdcM.Tag(0x0018,0x0091),
gdcM.Tag(0x0018,0x0093),
gdcM.Tag(0x0018,0x0094),
gdcM.Tag(0x0018,0x0095),
gdcM.Tag(0x0018,0x1088),
gdcM.Tag(0x0018,0x1090),
gdcM.Tag(0x0018,0x1094),
gdcM.Tag(0x0018,0x1250),
gdcM.Tag(0x0018,0x1251),
gdcM.Tag(0x0018,0x1310),
gdcM.Tag(0x0018,0x1312),
gdcM.Tag(0x0018,0x1314),
gdcM.Tag(0x0018,0x1315),
gdcM.Tag(0x0018,0x1316),
gdcM.Tag(0x0020,0x0110),
gdcM.Tag(0x0028,0x0120),
gdcM.Tag(0x0028,0x1050),
gdcM.Tag(0x0028,0x1051)
]
for tag in taglist:
    #print tag
    ano.Remove( tag )

# special handling
gen = gdcM.UIDGenerator()
ano.Replace( gdcM.Tag(0x0008,0x9123), gen.Generate() )
#ano.Empty( gdcM.Tag(0x0040,0x0555) )

#
# uid = gen.Generate()
# de.SetTag( gdcM.Tag(0x0008,0x0018) )
# de.SetByteValue( uid, gdcM.VL(len(uid)) )
# ds.Insert( de )

# init FMI now:
#fmi = f.GetHeader()
#ts = gdcM.TransferSyntax()
#print ts
#fmi.SetDataSetTransferSyntax( ts ) # default
#print fmi.GetDataSetTransferSyntax()
#de.SetTag( gdcM.Tag(0x0002,0x0010) )
#uid = "1.2.840.10008.1.2"
#de.SetByteValue( uid, gdcM.VL(len(uid)) )
#fmi.Insert( de )
# f.SetHeader( r.GetFile().GetHeader() )

writer = gdcM.Writer()
writer.SetFile( ano.GetFile() )
writer.SetFileName( "rawstorage.dcm" );
writer.Write()

```

27.22 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 (gdcmm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
*/
#include "gdcmmReader.h"
#include "gdcmmImageReader.h"
#include "gdcmmImageWriter.h"
#include "gdcmmCSAHeader.h"
#include "gdcmmAttribute.h"
#include "gdcmmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    // gdcmmDataExtra/gdcmmNonImageData/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];

    gdcmm::Reader reader; // Do not use ImageReader
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }

    gdcmm::CSAHeader csa;
    const gdcmm::DataSet& ds = reader.GetFile().GetDataSet
        ();

    const gdcmm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag
        ();
    //std::cout << t1 << std::endl;
    //const gdcmm::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 gdcmm::CSAElement &csael = csa.GetCSAElementByName
            ( "Columns" );
        std::cout << csael << std::endl;
        //const gdcmm::ByteValue *bv = csael.GetByteValue();
        gdcmm::Element<gdcmm::VR::IS, gdcmm::VM::VM1>
            el;
        el.Set( csael.GetValue() );
        dims[0] = el.GetValue();
        std::cout << "Columns:" << el.GetValue() << std::endl;
    }

    if( csa.FindCSAElementByName( "Rows" ) )
    {
        const gdcmm::CSAElement &csael2 = csa.GetCSAElementByName
            ( "Rows" );
        std::cout << csael2 << std::endl;
        gdcmm::Element<gdcmm::VR::IS, gdcmm::VM::VM1>
            el2;
        el2.Set( csael2.GetValue() );
        dims[1] = el2.GetValue();
        std::cout << "Rows:" << el2.GetValue() << std::endl;
    }

    double spacing[2] = { 1. , 1. };
    bool spacingfound = false;
    if( csa.FindCSAElementByName( "PixelSpacing" ) )
    {
        const gdcmm::CSAElement &csael3 = csa.GetCSAElementByName
            ( "PixelSpacing" );
        if( !csael3.IsEmpty() )
        {

```

```

        std::cout << csael3 << std::endl;
        gdcm::Element<gdcm::VR::DS, gdcm::VM::VM2>
            el3;
        el3.Set( csael3.GetValue() );
        spacing[0] = el3.GetValue(0);
        spacing[1] = el3.GetValue(1);
        std::cout << "PixelSpacing:" << el3.GetValue() << ", " << el3.
            GetValue(1) << std::endl;
        spacingfound = true;
    }
}

if( !spacingfound )
{
    std::cerr << "Problem with PixelSpacing" << std::endl;
    //return 1;
}
if( !dims[0] || !dims[1] )
{
    std::cerr << "Problem with dims" << std::endl;
    return 1;
}

gdcm::ImageWriter writer;

gdcm::Image &image = writer.GetImage();
image.SetNumberOfDimensions( 2 ); // good default
image.SetDimension(0, dims[0] );
image.SetDimension(1, dims[1] );
image.SetSpacing(0, spacing[0] );
image.SetSpacing(1, spacing[1] );
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() );
gdcm::PrivateTag csanonimaget(0x7fe1,0x10,"SIEMENS CSA
    NON-IMAGE");
const gdcm::DataElement &pixeldata = ds.GetDataElement
    ( csanonimaget );
image.SetDataElement( pixeldata );

std::string outfilename = "outcsa.dcm";
//writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

27.23 CStoreQtProgress.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

```

All rights reserved.
See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```
=====*/
/*
 * This small example show how one can use the virtual function
 * mechanism of the SimpleSubjectWatcher class to redirect progress
 * report to a custom Qt classes
 *
 * http://doc.qt.nokia.com/latest/qprogressdialog.html
 *
 * Usage:
 * CStoreQtProgress dicom.example.com 11112 gdcmData/
 *   MR_Spectroscopy_SIEMENS_OF.dcm
 *
 */

#include "gdcmServiceClassUser.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmProgressEvent.h"
#include "gdcmDirectory.h"
#include "gdcmPresentationContextGenerator.h"

#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>

namespace gdcm {
/*
 * This class is a little more complicated than what this example demonstrate
 * This watcher is capable of handling nested progress. Since the Progress
 * grows from [0 to 1] on a per file basis and we only have one instance of a
 * watcher per association, we need some calculation to compute the global
 * (total) progress
 * In fact we simply divide the per-file progress by the number of files.
 *
 * This QtWatcher class will then update the progress bar according to the
 * progress.
 */
class MyQtWatcher : public SimpleSubjectWatcher
{
    size_t nfiles;
    double progress;
    size_t index;
    double refprogress;
    QWidget* win;
    QProgressDialog* qtprogress;
public:
    MyQtWatcher(Subject * s, const char *comment = "", QWidget *w = NULL,
        QProgressDialog* p = NULL, size_t n = 1):
        SimpleSubjectWatcher(s,comment),nfiles(n),progress(0),index(0),refprogress(
            0),win(w),qtprogress(p) {}
    void ShowIteration()
    {
        index++;
        assert( index <= nfiles );
        // update refprogress (we are moving to the next file)
        refprogress = progress;
    }
    void ShowProgress(Subject *, const Event &evt)
    {
        // Retrieve the ProgressEvent:
        const ProgressEvent &pe = dynamic_cast<const ProgressEvent*>(evt);
        // compute global progress:
        progress = refprogress + (1. / 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 = progress * 100 + 0.5; // round to next int
        qtprogress->setValue(i);
        win->show();
    }
    virtual void ShowDataSet(Subject *caller, const Event &evt) {}
};
} // end namespace gdcm
```

```

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);

    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( portno );
    scu.SetTimeout( 1000 );
    scu.SetCalledAETitle( "GDCM_STORE" );

    if( !scu.InitializeConnection() )
    {
        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) )
    {
        return 1;
    }

    // Setup PresentationContext(s)
    scu.SetPresentationContexts( generator.
        GetPresentationContexts() );

    // Start ASSOCIATION
    if( !scu.StartAssociation() )
    {
        return 1;
    }

    // Send C-STORE
    if( !scu.SendStore( filename ) )
    {
        return 1;
    }

    // Stop ASSOCIATION
    if( !scu.StopAssociation() )
    {
        return 1;
    }

    win->show();

    return a.exec();
}

```

27.24 DecompressImage.cs

This is a C# example on how to use gdcm::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;
}

```

27.25 DecompressImage.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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:

    python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
"""

import gdcm
import sys

if __name__ == "__main__":

    file1 = sys.argv[1]
    file2 = sys.argv[2]

    r = gdcm.ImageReader()
    r.SetFileName( file1 )
    if not r.Read():
        sys.exit(1)

    image = gdcm.Image()
    ir = r.GetImage()

    image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
    dims = ir.GetDimensions();
    print ir.GetDimension(0);
    print ir.GetDimension(1);
    print "Dims:", dims

    # Just for fun:
    dircos = ir.GetDirectionCosines()
    t = gdcm.Orientation.GetType(dircos)
    l = gdcm.Orientation.GetLabel(t)
    print "Orientation label:", l

    image.SetDimension(0, ir.GetDimension(0) );
    image.SetDimension(1, ir.GetDimension(1) );

    pixeltype = ir.GetPixelFormat();
    image.SetPixelFormat( pixeltype );

    pi = ir.GetPhotometricInterpretation();
    image.SetPhotometricInterpretation( pi );

    pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010)
    )
    str1 = ir.GetBuffer()
    #print ir.GetBufferLength()
    pixeldata.SetByteValue( str1, gdcm.VL( len(str1) ) )
    image.SetDataElement( pixeldata )

    w = gdcm.ImageWriter()
    w.SetFileName( file2 )
    w.SetFile( r.GetFile() )
    w.SetImage( image )

```

```
if not w.Write():
    sys.exit(1)
```

27.26 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):
*
* $ gdcmrw -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
* gdcmrw 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;
}

```

27.27 DecompressJPEGFile.cs

This is a C# example on how to use `gdcm::SequenceOfFragments`

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre
 All rights reserved.
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 PURPOSE. See the above copyright notice for more information.

```

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressJPEGFile.exe somejpegfile.jpg
 */
using System;
using gdcm;

public class DecompressJPEGFile
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        System.IO.FileStream infile =
            new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.
                FileAccess.Read);
        uint fsize = gdcm.PosixEmulation.FileSize(file1);

        byte[] jstream = new byte[fsize];
        infile.Read(jstream, 0, jstream.Length);

        Trace.DebugOn();
        Image image = new Image();
        image.SetNumberOfDimensions( 2 ); // important for now
        DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

        // DO NOT set a ByteValue here, JPEG is a particular kind of encapsulated
        // syntax
        // in which can one cannot use a simple byte array for storage. Instead,
        // see
        // gdcm.SequenceOfFragments
        //pixeldata.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );

        // Create a new SequenceOfFragments C++ object, store it as a SmartPointer
        :
        SmartPtrFrag sq = SequenceOfFragments.New();
        Fragment frag = new Fragment();
        frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );
        // Single file => single fragment
        sq.AddFragment( frag );
        // Pass by reference:
        pixeldata.SetValue( sq.__ref__() );

        // insert:
        image.SetDataElement( pixeldata );

        // JPEG use YBR to achieve better compression ratio by default (not RGB)
        // FIXME hardcoded:
        PhotometricInterpretation pi = new PhotometricInterpretation(
            PhotometricInterpretation.PIType.YBR_FULLL );
        image.SetPhotometricInterpretation( pi );
        // FIXME hardcoded:
        PixelFormat pixeltype = new PixelFormat(3,8,8,7);
        image.SetPixelFormat( pixeltype );

        // FIXME hardcoded:
        image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.
            JPEGLosslessProcess14_1 ) );
        image.SetDimension(0, 692);
        image.SetDimension(1, 721);

        // Decompress !
        byte[] decompressedData = new byte[(int)image.GetBufferLength()];
        image.GetBuffer(decompressedData);

        // Write out the decompressed bytes
        System.Console.WriteLine(image.toString());
        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/dd.raw",
                System.IO.FileMode.Create))
        {

```

```

        System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
        writer.Write(decompressedData);
    }

    return 0;
}
}

```

27.28 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();
        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.GetPixmap() );
        if( !change.Change() )
        {
            throw new Exception("Could not change: " + file1 );
        }

        // When using a PixmapReader the following code crashes, I do not
        // understand why (MM)
        // Instead hack our way in, and use an ImageReader instead of a
        // PixmapReader
        //
        // Hum looks like Java Covariant Return type is not working for some reason
        // Pixmap out = ((PixmapToPixmapFilter)change).GetOutput(); // old syntax
        Pixmap out2 = (Pixmap)change.GetOutput(); // new syntax
        System.out.println( out2.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( out );
    writer.SetImage( out2 );
    ret = writer.Write();
    if( !ret )
    {
        throw new Exception("Could not write: " + file2 );
    }
}
}

```

27.29 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;
}

```

27.30 DiscriminateVolume.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 "gdcmlScanner.h"
#include "gdcmlTesting.h"
#include "gdcmlIPPSorter.h"
#include "gdcmlDirectionCosines.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 gdcml {
    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::FilenameType::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 & s, Directory::FileNamesType
    const & subset, const char *iopval)
{
    std::cout << "IOP: " << iopval << std::endl;
    IPPSorter ipp;
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 ); // ??
    bool b = ipp.Sort( subset );
    if( !b )
    {
        // If you reach here this means you need one more parameter to discriminat
        this
        // series. Eg. T1 / T2 intertwinced. 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(), gdc::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);

    gdc::Scanner::ValueType vt3 = s.GetValues(t3);
    for(
        gdc::Scanner::ValueType::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;
    gdc::Scanner::ValueType vt2 = s.GetValues(t2);
    for(
        gdc::Scanner::ValueType::const_iterator it = vt2.begin()
        ; it != vt2.end(); ++it )
    {
        ProcessASeries(s, it->c_str());
    }
}

public:

void Print( std::ostream & os )
{
    os << "Sorted Files: " << std::endl;
    for(
        std::vector< Directory::FileNamesType >::const_iterator it = SortedFiles.
            begin();
        it != SortedFiles.end(); ++it )
    {
        os << "Group: " << std::endl;
        for(
            Directory::FileNamesType::const_iterator file = it->begin();
            file != it->end(); ++file)
        {
            os << *file << std::endl;
        }
    }
    os << "Unsorted Files: " << std::endl;
    for(
        std::vector< Directory::FileNamesType >::const_iterator it = UnsortedFiles.
            begin();
        it != UnsortedFiles.end(); ++it )
    {

```

```

    os << "Group: " << std::endl;
    for(
        Directory::FileNamesType::const_iterator file = it->begin();
        file != it->end(); ++file)
    {
        os << *file << std::endl;
    }
}

std::vector< Directory::FileNamesType > const & GetSortedFiles() const {
    return SortedFiles; }
std::vector< Directory::FileNamesType > const & GetUnsortedFiles() const {
    return UnsortedFiles; }

void ProcessIntoVolume( Scanner const & s )
{
    gdcmm::Scanner::ValuesType vt1 = s.GetValues(
        gdcmm::t1 );
    for(
        gdcmm::Scanner::ValuesType::const_iterator it = vt1.begin()
        ; it != vt1.end(); ++it )
    {
        ProcessAStudy( s, it->c_str() );
    }
}

};

} // namespace gdcmm

int main(int argc, char *argv[])
{
    std::string dirl;
    if( argc < 2 )
    {
        const char *extradataroot = NULL;
#ifdef GDCM_BUILD_TESTING
        extradataroot = gdcmm::Testing::GetDataExtraRoot
            ();
#endif
        if( !extradataroot )
        {
            return 1;
        }
        dirl = extradataroot;
        dirl += "/gdcmmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
    else
    {
        dirl = argv[1];
    }

    gdcmm::Directory d;
    d.Load( dirl.c_str(), true ); // recursive !

    gdcmm::Scanner s;
    s.AddTag( gdcmm::t1 );
    s.AddTag( gdcmm::t2 );
    s.AddTag( gdcmm::t3 );
    s.AddTag( gdcmm::t4 );
    bool b = s.Scan( d.GetFileNames() );
    if( !b )
    {
        std::cerr << "Scanner failed" << std::endl;
        return 1;
    }

    gdcmm::DiscriminateVolume dv;
    dv.ProcessIntoVolume( s );
    dv.Print( std::cout );

    return 0;
}

```

27.31 DumbAnonymizer.py

```
#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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 can use the gdcm.Anonymizer in 'dumb' mode.
This class becomes really handy when one knows which particular tag to fill in.

Usage:

python DumbAnonymizer.py gdcmData/012345.002.050.dcm out.dcm

"""

import gdcm

# http://www.oid-info.com/get/1.3.6.1.4.17434
THERALYS_ORG_ROOT = "1.3.6.1.4.17434"

tag_rules={
    # Value
    (0x0012,0x0010):("Value","MySponsorName"),
    (0x0012,0x0020):("Value","MyProtocolID"),
    (0x0012,0x0021):("Value","MyProtocolName"),
    (0x0012,0x0062):("Value","YES"),
    (0x0012,0x0063):("Value","MyDeidentificationMethod"),

    # Method
    (0x0002,0x0003):("Method","GenerateMSOPIId"),
    (0x0008,0x1155):("Method","GenerateMSOPIId"),
    (0x0008,0x0018):("Method","GenerateMSOPIId"),
    (0x0010,0x0010):("Method","GetSponsorInitials"),
    (0x0010,0x0020):("Method","GetSponsorId"),
    (0x0012,0x0030):("Method","GetSiteId"),
    (0x0012,0x0031):("Method","GetSiteName"),
    (0x0012,0x0040):("Method","GetSponsorId"),
    (0x0012,0x0050):("Method","GetTPIId"),
    (0x0018,0x0022):("Method","KeepIfExist"),
    (0x0018,0x1315):("Method","KeepIfExist"),
    (0x0020,0x000d):("Method","GenerateStudyId"),
    (0x0020,0x000e):("Method","GenerateSeriesId"),
    (0x0020,0x1002):("Method","GetNumberOfFrames"),
    (0x0020,0x0020):("Method","GetPatientOrientation"),
    # Other:
    (0x0012,0x0051):("Patient Field","Type Examen"),
    (0x0018,0x1250):("Sequence Field","Receive Coil"),
    (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
    (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
    (0x0018,0x0082):("Sequence Field","Inversion Time"),
}

class MyAnon:
    def __init__(self):
        self.studyuid = None
        self.seriesuid = None
        generator = gdcm.UIDGenerator()
        if not self.studyuid:
            self.studyuid = generator.Generate()
        if not self.seriesuid:
            self.seriesuid = generator.Generate()
    def GetSponsorInitials(self):
        return "dummy^foobar"
    def GenerateStudyId(self):
        return self.studyuid
    def GenerateSeriesId(self):
        return self.seriesuid
    #def GenerateMSOPIId(self):
    def GenerateMSOPIId(self):
```

```

        generator = gdcm.UIDGenerator()
        return generator.Generate()
def GetSiteId(self):
    return "MySiteId"
def GetSiteName(self):
    return "MySiteName"
def GetSponsorId(self):
    return "MySponsorId"
def GetTPId(self):
    return "MyTP"

if __name__ == "__main__":
    import sys
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle
        ( "DumbAnonymizer" )
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )

    r = gdcm.Reader()
    filename = sys.argv[1]
    r.SetFileName( filename )
    if not r.Read(): sys.exit(1)

    obj = MyAnon()

    w = gdcm.Writer()
    ano = gdcm.Anonymizer()
    ano.SetFile( r.GetFile() )
    ano.RemoveGroupLength()
    for tag,rule in tag_rules.items():
        if rule[0] == 'Value':
            print tag,rule
            ano.Replace( gdcm.Tag( tag[0], tag[1] ), rule[1] )
        elif rule[0] == 'Method':
            print tag,rule
            # result = locals()[rule[1]]()
            methodname = rule[1]
            if hasattr(obj, methodname):
                _member = getattr(obj, methodname)
                result = _member()
                ano.Replace( gdcm.Tag( tag[0], tag[1] ), result )
            else:
                print "Problem with: ", methodname

    outfilename = sys.argv[2]
    w.SetFileName( outfilename )
    w.SetFile( ano.GetFile() )
    if not w.Write(): sys.exit(1)

```

27.32 DumpADAC.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * the goal of this example is to mimic the behavior of disp_img_header
 * see http://www.gmecorp-usa.com/IM/NM/GC/ADAC/SV/adactechtips/
   Released_01Q3.pdf
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>
#include <vector>

```

```

#include <string.h>
#include <assert.h>
#include <stdint.h>

struct dict
{
    uint16_t key;
    const char *name;
};

dict Array[] = {
    { 0x01, "Patient name" },
    { 0x02, "Patient ID" },
    { 0x03, "Patient sex" },
    { 0x04, "Patient age" },
    { 0x05, "Patient height" },
    { 0x06, "Patient weight" },
    { 0x07, "Exam date" },
    { 0x08, "Dose admin. time" },
    { 0x09, "Unique exam key" },
    { 0x0a, "Exam procedure" },
    { 0x0b, "Referring physician" },
    { 0x0c, "Attending physician" },
    { 0x0d, "Imaging modality" },
    { 0x0e, "Hospital ID" },
    { 0x0f, "Histogram crv file" },
    { 0x10, "Acq. start time" },
    { 0x11, "Object data type" },
    { 0x12, "Image viewid" },
    { 0x13, "Imaging device name" },
    { 0x14, "Device serial number" },
    { 0x15, "Collimator" },
    { 0x16, "Software version" },
    { 0x17, "Radiopharmaceutical #1" },
    { 0x18, "Energy window #1 center" },
    { 0x19, "Radiopharmaceutical #2" },
    { 0x1a, "Energy window #1 width" },
    { 0x1b, "Isotope imaging mode" },
    { 0x1c, "Energy window #2 center" },
    { 0x1d, "Energy window #2 width" },
    { 0x1e, "Energy window #3 center" },
    { 0x1f, "Energy window #3 width" },
    { 0x20, "Energy window #4 center" },
    { 0x21, "Energy window #4 width" },
    { 0x22, "??Energy window #5 center" },
    { 0x23, "??Energy window #5 width" },
    { 0x24, "Patient orientation" },
    { 0x25, "Spatial resolution" },
    { 0x26, "Slice thickness" },
    { 0x27, "Image X dimension" },
    { 0x28, "Image Y dimension" },
    { 0x29, "Image Z dimension" },
    { 0x2a, "Image pixel width" },
    { 0x2b, "Uniformity corr. file" },
    { 0x2c, "Acquisition zoom factor" },
    { 0x2d, "Total counts in set" },
    { 0x2e, "Time / frame" },
    { 0x2f, "Total acq. time" },
    { 0x30, "Maximum pixel value" },
    { 0x31, "Minimum pixel value" },
    { 0x32, "R-R interval time" },
    { 0x33, "Percent of cycle imaged" },
    { 0x34, "# of cycles accepted" },
    { 0x35, "# of cycles rejected" },
    { 0x36, "Approximate ED frame" },
    { 0x37, "Approximate ES frame" },
    { 0x38, "Approximate EF" },
    { 0x39, "Starting angle" },
    { 0x3a, "Degrees of rotation" },
    { 0x3b, "Direction of rotation" },
    { 0x3c, "Cont. or step/shoot" },
    { 0x3d, "Lim recon start frame" },
    { 0x3e, "Upper window grey shade" },
    { 0x3f, "Lower lvl grey shade" },
    { 0x40, "Associated color map" },
    { 0x41, "Custom color map file" },
    { 0x42, "Manipulated image" },
    { 0x43, "Axis of rotation corr." },
    { 0x44, "Reorientation azimuth" },
    { 0x45, "Reorientation elevation" },
    { 0x46, "Filter type" },

```

```

    { 0x47, "Filter order" },
    { 0x48, "Filter cutoff frequency" },
    { 0x49, "Reconstruction type" },
    { 0x4a, "Attenuation coefficient" },
    { 0x4b, "Associated parent file" },
    { 0x4c, "Unique patient key" },
    { 0x52, "Normalization crv file" },
    { 0x53, "Unique object key" },
    { 0x54, "This phase of VFR is" },
    { 0x55, "True color value" },
    { 0x56, "# of sets of x,y,z grps" },
    { 0x57, "Scale factor of set" },
    { 0x6d, "Date of birth" },
    { 0x6e, "Directional orientation" },
    { 0x6f, "Number of VFR studies" },
    { 0x70, "R-R low tolerance" },
    { 0x71, "R-R high tolerance" },
    { 0x72, "Prog specific results:" },

    { 0x99, 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 (val>>8) | (val<<8);
}

uint32_t readint32(std::istream &is )
{
    uint32_t val;
    is.read( (char*)&val, sizeof( val ));
    val= ((val<<8)&0xFF00FF00) | ((val>>8)&0x00FF00FF);
    return (val>>16) | (val<<16);
}

float readfloat32(std::istream &is )
{
    union { uint32_t val; float f;} dual;
    dual.val = readint32(is);
    return dual.f;
}

struct el

```

```

{
    uint16_t v1;
    uint16_t v2;
    uint16_t v3;
    void read( std::istream & is )
    {
        v1 = readint16(is);
        v2 = readint16(is);
        v3 = readint16(is);
    }
    void print( std::ostream & os )
    {
        os << std::hex << v1 << "\\t" << v2 << "\\t" << v3 << std::endl;
    }
};

std::vector<el> Vel;

void readelement( std::istream & is )
{
    el e;
    e.read( is );
    Vel.push_back( e );
}

void printascii( uint16_t tag, const char *buffer, size_t len )
{
    std::ostream & os = std::cout;
    if( tag == 0x72 )
    {
        os << "\\n ";
        for( size_t i = 0; i < len; ++i )
        {
            const char &c = buffer[i];
            if( c == 0x0 ) os << "!";
            else if( c == 0x0f ) os << " ";
            else if( c == 0x17 ) os << ":";
            else if( c == 0x14 ) os << ":";
            else if( c == 0x10 ) os << ":";
            else if( c == 0x16 ) os << ":";
            else if( c == 0x08 ) os << ":";
            else if( c == 0x0b ) os << ":";
            else if( c == 0x0e ) os << ":";
            else if( c == 0x07 ) os << ":";
            else os << c;
        }
        os << " ";
    }
    else
    {
        (void)len;
        os << " " << buffer << " ";
    }
}

bool DumpADAC( std::istream & is )
{
    std::ostream &os = std::cout;

    char magic[6 + 1];
    magic[6] = 0;
    is.read( magic, 6);
    // std::cout << magic << " ";
    assert( strcmp( magic, "adac01" ) == 0 );
    char c = is.get();
    assert( c == 0 );
    c = is.get();
    assert( c == 'X' );

    uint16_t v;
    v = readint16(is);
    // std::cout << v << std::endl;
    assert( v == 512 ); // ??

    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 gdcmm::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::ostringstream is;
std::string dup( bv->GetPointer(), bv->GetLength() );
is.str( dup );
bool b = DumpADAC( is );
if( !b ) return 1;

return 0;
}

```

27.33 DumpGEMSMovieGroup.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 "gdcmmImage.h"
#include "gdcmmImageWriter.h"
#include "gdcmmDataElement.h"
#include "gdcmmPrivateTag.h"
#include "gdcmmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

bool PrintNameValuePairMapping( gdcmm::SequenceOfItems *
    sqi_values,
gdcmm::SequenceOfItems *sqi_names, std::string const &
    indent )
{
    using namespace gdcmm;
    // prepare names mapping:
    typedef VRToType<VR::UL>::Type UL;
    std::map< UL, std::string > names;
    assert( sqi_names );
    assert( sqi_values );
    SequenceOfItems::SizeType s = sqi_names->
        GetNumberOfItems();
    PrivateTag tindex(0x7fe1,0x71,"GEMS_Ultrasound_MovieGroup_001");
    PrivateTag tname (0x7fe1,0x72,"GEMS_Ultrasound_MovieGroup_001");
    // First sequence contains all possible names (this is a dict)
    for( SequenceOfItems::SizeType i = 1; i <= s; ++i )
    {
        const Item & item = sqi_names->GetItem( i );
    }
}

```

```

const DataSet & ds = item.GetNestedDataSet();
if( !ds.FindDataElement( tindex )
    || !ds.FindDataElement( tname ) )
{
    return false;
}
const DataElement & index = ds.GetDataElement(
    tindex );
const DataElement & name = ds.GetDataElement(
    tname );
if( index.IsEmpty() || name.IsEmpty() )
{
    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< UL, std::string > ( 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 ) )
    {
        return false;
    }
    const DataElement & index2 = ds.GetDataElement(
        tindex2 );
    if( index2.IsEmpty() )
    {
        return false;
    }
    gdcm::Element<VR::FD, VM::VM1_2> el1;
    el1.SetFromDataElement( index2 );

    UL copy = el1.GetValue();
    #if 1
        std::cout << indent;
        std::cout << " ( " << names[ copy ];
    #endif
    // (7fe1,1052) FD 1560 # 8,1 ?
    // (7fe1,1057) LT [MscSkelSup] # 10,1 ?
    //PrivateTag tvalue(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001");
    PrivateTag tvalueint(0x7fe1,0x49,"GEMS_Ultrasound_MovieGroup_001"
    ); // UL
    PrivateTag tvaluefloat1(0x7fe1,0x51,"
    GEMS_Ultrasound_MovieGroup_001"); // FL
    PrivateTag tvaluefloat(0x7fe1,0x52,"
    GEMS_Ultrasound_MovieGroup_001"); // FD
    PrivateTag tvalueul(0x7fe1,0x53,"GEMS_Ultrasound_MovieGroup_001"
    ); // UL
    PrivateTag tvaluesl(0x7fe1,0x54,"GEMS_Ultrasound_MovieGroup_001"
    ); // SL
    PrivateTag tvalueob(0x7fe1,0x55,"GEMS_Ultrasound_MovieGroup_001"
    ); // OB
    PrivateTag tvaluetext(0x7fe1,0x57,"GEMS_Ultrasound_MovieGroup_001
    "); // LT
    PrivateTag tvaluefd(0x7fe1,0x77,"GEMS_Ultrasound_MovieGroup_001"
    ); // FD / 1-N
    PrivateTag tvaluesl3(0x7fe1,0x79,"GEMS_Ultrasound_MovieGroup_001"
    ); // SL / 1-N
    PrivateTag tvaluesl2(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001"
    ); // SL ??
    PrivateTag tvaluefd1(0x7fe1,0x87,"GEMS_Ultrasound_MovieGroup_001"
    ); // FD / 1-N
    PrivateTag tvaluefloat2(0x7fe1,0x88,"
    GEMS_Ultrasound_MovieGroup_001"); // FD ??
    #if 1
        std::cout << " ) = ";
    #endif
    if( ds.FindDataElement( tvalueint ) )

```

```

    {
        const DataElement & value = ds.GetDataElement(
            tvalueint );
        gdcm::Element<VR::UL,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat1 ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluefloat1 );
        gdcm::Element<VR::FL,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluefloat );
        gdcm::Element<VR::FD,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluesl ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluesl );
        gdcm::Element<VR::SL,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvalueul ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvalueul );
        gdcm::Element<VR::UL,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        assert( el2.GetLength() == 1 );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvalueob ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvalueob );
        gdcm::Element<VR::SL,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
        std::cout << value << std::endl;
    }
    else if( ds.FindDataElement( tvaluetext ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluetext );
        gdcm::Element<VR::LT,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvaluesl2 ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluesl2 );
        gdcm::Element<VR::SL,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        assert( el2.GetLength() == 4 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluesl3 ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluesl3 );
        gdcm::Element<VR::SL,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        assert( el2.GetLength() == 4 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluefd ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluefd );

```

```

        gdcmm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        el2.Print( std::cout );
//      assert( el2.GetLength() == 4 || el2.GetLength() == 3 || el2.GetLength()
//      == 8 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat2 ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluefloat2 );
        gdcmm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        assert( el2.GetLength() == 2 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluefd1 ) )
    {
        const DataElement & value = ds.GetDataElement(
            tvaluefd1 );
        gdcmm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        assert( el2.GetLength() == 4 );
        std::cout << std::endl;
    }
    else
    {
        std::cout << "(no value)" << std::endl;
//      std::cout << ds << std::endl;
        assert( ds.Size() == 2 );
    }
}
return true;
}

bool PrintNameValueMapping2( gdcmm::PrivateTag const & privtag,
    const gdcmm::DataSet & ds,
    gdcmm::SequenceOfItems *sqi_names, std::string const &
    indent )
{
    if( !ds.FindDataElement( privtag ) ) return 1;
    const gdcmm::DataElement& seq_values = ds.GetDataElement
        ( privtag );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems>
        sqi = seq_values.GetValueAsSQ();

    return PrintNameValueMapping( sqi, sqi_names, indent);
}

bool PrintNameValueMapping3( gdcmm::PrivateTag const & privtag1,
    gdcmm::PrivateTag const & privtag2, const gdcmm::DataSet
    & ds,
    gdcmm::SequenceOfItems *sqi_names, std::string const &
    indent )
{
    if( !ds.FindDataElement( privtag1 ) )
    {
        assert( 0 );
        return false;
    }
    const gdcmm::DataElement& values10name = ds.GetDataElement
        ( privtag1 );
    gdcmm::Element<gdcmm::VR::LO,gdcmm::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( gdcmm::DataSet const & ds10, gdcmm::SequenceOfItems
    *sqi_dict, std::string const & indent )
{
    const gdcmm::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();

    int ni3 = sqi_values73->GetNumberOfItems();
    for( int 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(0x7fel,0x74,"
            GEMS_Ultrasound_MovieGroup_001");
        const gdcm::PrivateTag tseq_values75(0x7fel,0x75,"
            GEMS_Ultrasound_MovieGroup_001");
        PrintNameValueMapping3( tseq_values74name, tseq_values75, ds73, 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(0x7fel,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();

    int ni3 = sqi_values83->GetNumberOfItems();
    for( int 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(0x7fel,0x84,"
            GEMS_Ultrasound_MovieGroup_001");
        const gdcm::PrivateTag tseq_values85(0x7fel,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 )
{
    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();

    int nil = sqi_values10->GetNumberOfItems();
    // assert( nil == 1 );
    for( int i1 = 1; i1 <= nil; ++i1 )
    {
        gdcm::Item &item_10 = sqi_values10->GetItem(i1);

```

```

gdcmm::DataSet &ds10 = item_10.GetNestedDataSet
();
assert( ds10.Size() == 2 + 3 );
// (7fel,0010)
// (7fel,1012)
// (7fel,1018)
// (7fel,1020)
// (7fel,1083)

PrintNameValueMapping3( privtag1, privtag2, ds10, sqi_dict, " " );
std::cout << std::endl;

const gdcmm::PrivateTag tseq_values20(0x7fel,0x20,"
GEMS_Ultrasound_MovieGroup_001");
if( !ds10.FindDataElement( tseq_values20 ) )
{
    assert( 0 );
    return 1;
}
const gdcmm::DataElement& seq_values20 = ds10.
GetDataElement( tseq_values20 );
gdcmm::SmartPointer<gdcmm::SequenceOfItems>
sqi_values20 = seq_values20.GetValueAsSQ();

int ni2 = sqi_values20->GetNumberOfItems();
//assert( ni == 1 );
for( int i2 = 1; i2 <= ni2; ++i2 )
{
    gdcmm::Item &item_20 = sqi_values20->GetItem(i2);
    gdcmm::DataSet &ds20 = item_20.GetNestedDataSet
();
    int count = ds20.Size();
    assert( ds20.Size() == 2 + 3 || ds20.Size() == 2 + 2 );
    // (7fel,0010)
    // (7fel,1024)
    // (7fel,1026)
    // (7fel,1036)
    // (7fel,1083) (*)

    const gdcmm::PrivateTag tseq_values20name(0x7fel,0x24,"
GEMS_Ultrasound_MovieGroup_001");
    const gdcmm::PrivateTag tseq_values26(0x7fel,0x26,"
GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values20name, tseq_values26, ds20, sqi_dict,
        " " );
    std::cout << std::endl;

    print83(ds20, sqi_dict, " ");
}

print83(ds10, sqi_dict, " ");
}
return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcmm;
    const char *filename = argv[1];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    reader.Read();

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    const PrivateTag tseq(0x7fel,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;
    const DataElement& seq = ds.GetDataElement( tseq );

    SmartPointer<SequenceOfItems> sqi = seq.
        GetValueAsSQ();
    assert( sqi->GetNumberOfItems() == 1 );

    Item &item = sqi->GetItem(1);
    DataSet &subds = item.GetNestedDataSet();

    const PrivateTag tseq_dict(0x7fel,0x70,"
GEMS_Ultrasound_MovieGroup_001");
    if( !subds.FindDataElement( tseq_dict ) ) return 1;

```

```

const DataElement& seq_dict = subds.GetDataElement(
    tseq_dict );
SmartPointer<SequenceOfItems> sqi_dict =
    seq_dict.GetValueAsSQ();

const PrivateTag tseq_values8(0x7fel,0x8,"
    GEMS_Ultrasound_MovieGroup_001");
if( !subds.FindDataElement( tseq_values8 ) ) return 1;
const DataElement& seq_values8 = subds.GetDataElement
    ( tseq_values8 );
SmartPointer<SequenceOfItems> sqi_values8 =
    seq_values8.GetValueAsSQ();

const PrivateTag tseq_values8name(0x7fel,0x2,"
    GEMS_Ultrasound_MovieGroup_001");
if( !subds.FindDataElement( tseq_values8name ) ) return 1;
const DataElement& values8name = subds.GetDataElement
    ( tseq_values8name );
{
    Element<VR::LO,VM::VM1> el;
    el.SetFromDataElement( values8name );
    std::cout << el.GetValue() << std::endl;
}
int count = subds.Size();
assert( subds.Size() == 3 + 2 + 1 || subds.Size() == 3 + 2 + 2);

// (7fel,0010) # 30,1 Private Creator
// (7fel,1002) # 8,1 US MovieGroup Value 0008 Name
// (7fel,1003) # 4,1 ?
// (7fel,1008) # 8140,1 US MovieGroup Value 0008 Sequence
// (7fel,1010) # 1372196,1 ?
// (7fel,1070) # 33684,1 US MovieGroup Dict
// (7fel,1073) (*)
PrintNameValueMapping( sqi_values8, sqi_dict, " ");

const PrivateTag tseq_values10(0x7fel,0x10,"
    GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq_values10name(0x7fel,0x12,"
    GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq_values18(0x7fel,0x18,"
    GEMS_Ultrasound_MovieGroup_001");
PrintNameValueMapping4( tseq_values10, subds, tseq_values10name,
    tseq_values18, sqi_dict," ");

print73( subds, sqi_dict, " " );

#if 0
    gdc::DataSet::ConstIterator it = subds.Begin
        ();
    for( ; it != subds.End(); ++it )
    {
        const gdc::DataElement &de = *it;
        std::cout << de.GetTag() << std::endl;
    }
#endif

return 0;
}

```

27.34 DumpToSQLITE3.cxx

```

/*=====
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.

=====*/
/*
* 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 "gdcmsScanner.h"
#include "gdcmsDirectory.h"
#include "gdcmsTag.h"
#include "gdcmsTrace.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);

    gdcms::Trace::SetDebug( false );
    gdcms::Trace::SetWarning( false );
    const char *inputdirectory = argv[1];

    gdcms::Directory d;
    unsigned int nfiles = d.Load( inputdirectory, true);

    gdcms::Scanner s;
    using gdcms::Tag;
    s.AddTag( Tag(0x20,0xd) ); // Study Instance UID
    s.AddTag( Tag(0x20,0xe) ); // Series Instance UID

    bool b = s.Scan( d.GetFilenames() );
    if( !b ) 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 gdcms::Directory;
    using gdcms::Scanner;
    const Directory::FilenamesType& files = d.GetFilenames();
    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,
                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;
}

```

27.35 DuplicatePCDE.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
/*

```

Usage:
DuplicatePCDE gdcmData/D_CLUNIE_CT1_J2KI.dcm out.dcm

aka:
medical.nema.org/medical/dicom/DataSets/WG04/IMAGES/J2KI/CT1_J2KI

See:
gdcmConformanceTests/CT1_J2KI_DuplicatePCDE.dcm

Original thread can be found at:

http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/82f28c4db28963af

Question:

1.
There is no restriction for a specific Private Creator Data Element (PCDE) to be unique within the same group, right ?
Decoders of Private Data would have to handle the case where a PCDE would be repeated and should NOT stop on the first instance of a particular PCDE, right ?

Eg. when searching for the tag associated with
(0x0029,0x0010,"SIEMENS CSA HEADER") in the following (pseudo) dataset:

```
(0029,0010) LO [SIEMENS CSA HEADER] # 18, 1
PrivateCreator
(0029,0011) LO [SIEMENS MEDCOM HEADER] # 22, 1
PrivateCreator
(0029,0012) LO [SIEMENS MEDCOM HEADER2] # 22, 1
PrivateCreator
(0029,0013) LO [SIEMENS CSA HEADER] # 18, 1
PrivateCreator
(0029,1008) CS [IMAGE NUM 4] # 12, 1
CSAImageHeaderType
(0029,1009) LO [20050723] # 8, 1
CSAImageHeaderVersion
(0029,1010) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
(0029,1018) CS [MR] # 2, 1
CSASeriesHeaderType
(0029,1019) LO [20050723] # 8, 1
CSASeriesHeaderVersion
(0029,1020) OB 53\56\31\30\04\03\02\01\2c\00\00\00\4d
\00\00\00\55\73\65\64\50\61... # 51520, 1 CSASeriesHeaderInfo
(0029,1131) LO [4.0.163088300] # 14, 1
PMTFInformation1
(0029,1132) UL 32768 # 4, 1
PMTFInformation2
(0029,1133) UL 0 # 4, 1
PMTFInformation3
(0029,1134) CS [DB TO DICOM] # 12, 1
PMTFInformation4
(0029,1260) ?? 63\6f\6d\20 # 4, 1
Unknown Tag & Data
(0029,1310) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
```

one should return two instances, correct ?

Answer:

I would say that this is covered in principle by the PS 3.5 7.1
"The Data Elements ... shall occur at most once in a Data Set"
rule, since the data element is defined by the tuple
(private creator,gggg,ee) where xxee is the element
number and xx is arbitrary and has no inherent meaning and
does not serve to disambiguate the data element.

E.g.:

```
(0019,0030) Private Creator ID = "Smith"
...
(0019,0032) Private Creator ID = "Smith"
...
(0019,3015) Fractal Index = "32"
...
(0019,3215) Fractal Index = "32"
```

would be illegal because even though they are assigned different

```

(completely arbitrary) blocks, with the same group, element
number and private creator, (0019,3015) and (0019,3215) are the
"same" data element.

*/

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Let's get all private element from group 0x9:
    /*
(0009,0010) LO [GEMS_IDEN_01] # 12,1
    Private Creator
(0009,1001) LO [GE_GENESIS_FF ] # 14,1 Full
    fidelity
(0009,1002) SH [CT01] # 4,1 Suite
    id
(0009,1004) SH [HiSpeed CT/i] # 12,1
    Product id
(0009,1027) SL 862399669 # 4,1 Image
    actual date
(0009,1030) SH (no value) # 0,1 Service
    id
(0009,1031) SH (no value) # 0,1 Mobile
    location number
(0009,10e6) SH [05] # 2,1 Genesis
    Version - now
(0009,10e7) UL 973283917 # 4,1 Exam
    Record checksum
(0009,10e9) SL 862399669 # 4,1 Actual
    series data time stamp
    */

    gdcm::Tag start(0x0009,0x0);
    // Create a temporary duplicate dataset, since we cannot insert data element
    // as we go over them (std::set
    // would reorganize itself as we go over it ...)
    gdcm::DataSet dup;
    gdcm::Tag new_private(0x0009,0x0);
    while (start.GetGroup() == 0x9 )
    {
        const gdcm::DataElement& de = ds.FindNextDataElement
            (start);
        const gdcm::Tag &t = de.GetTag();
        if( t.IsPrivateCreator() )
        {
            std::cout << t << std::endl;
            // Ok let's duplicate into the next available attribute:
            gdcm::DataElement duplicate = de;
            duplicate.GetTag().SetElement( 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:
        gdc::ByteValue *dupbv = new gdc::ByteValue
    ( bv->GetPointer(),
      bv->GetLength() );
    // Let's recognize the duplicated ASCII-type elements:
    if( duplicate.GetVR() & gdc::VR::VRASCII )
        dupbv->Fill( 'X' );
    duplicate.SetValue( *dupbv );
    }
    dup.Insert( duplicate );
    }
    start = t;
    // move to next possible 'public' element
    start.SetElement( start.GetElement() + 1 );
    }

    gdc::DataSet::ConstIterator it = dup.Begin(
    );
    for( ; it != dup.End(); ++it )
    {
        ds.Insert( *it );
    }

    gdc::Writer w;
    w.SetFile( file );
    w.SetFileName( outfilename );
    if ( !w.Write() )
    {
        return 1;
    }

    return 0;
}

```

27.36 ELSCINT1WaveToText.cxx

```

/*=====
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.

=====*/
#include "gdcReader.h"
#include "gdcPrivateTag.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 (gdc-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;
    int 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 (int 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' << " " << '\\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];

```

```

gdcmm::Reader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
}
const gdcmm::DataSet& ds = reader.GetFile().GetDataSet
();

const gdcmm::PrivateTag twave(0x01e1,0x18,"ELSCINT1");
if( !ds.FindDataElement( twave ) ) return 1;
const gdcmm::DataElement& wave = ds.GetDataElement
( twave );
if ( wave.IsEmpty() ) return 1;
const gdcmm::ByteValue * bv = wave.GetByteValue();
assert( bv );

std::ofstream os( outfilename );
// Dump that to a CSV file:
wave2stream( os, bv->GetPointer(), bv->GetLength() );
os.close();

return 0;
}

```

27.37 EncapsulateFileInRawData.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 "gdcmmAnonymizer.h"
#include "gdcmmWriter.h"
#include "gdcmmUIDGenerator.h"
#include "gdcmmFile.h"
#include "gdcmmTag.h"
#include "gdcmmSystem.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/gdcmm-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( !gdcmm::System::FileExists( filename ) ) return 1;

```

```

size_t s = gdcmm::System::FileSize(filename);

magic_t cookie = magic_open(MAGIC_NONE);
const char * file_type = magic_file(cookie, filename);
magic_close(cookie);

gdcmm::Writer w;
gdcmm::File &file = w.GetFile();
gdcmm::DataSet &ds = file.GetDataSet();
//w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );

file.GetHeader().SetDataSetTransferSyntax(
    gdcmm::TransferSyntax::ImplicitVRLittleEndian
);

gdcmm::Anonymizer anon;
anon.SetFile( file );

gdcmm::MediaStorage ms = gdcmm::MediaStorage::RawDataStorage
;

gdcmm::UIDGenerator gen;
anon.Replace( gdcmm::Tag(0x0008,0x16), ms.GetString()
);
std::cout << ms.GetString() << std::endl;
anon.Replace( gdcmm::Tag(0x0008,0x18), gen.Generate()
);

if (!w.Write() )
{
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

27.38 ExtractEncapsulatedFile.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.

=====*/

/*
* 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
* MIMETimeTypeOfEncapsulatedDocument
* ...
*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcmm/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;
    }
}

```

27.39 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 &nesteddds = item.GetNestedDataSet(
        );

    if( ! nesteddds.FindDataElement( gdcm::Tag( 0x0400,0
        x0520) ) ) return 1;

    const gdcm::DataElement &EncryptedContent = nesteddds.
        GetDataElement( gdcm::Tag( 0x0400,0x0520) );

    const gdcm::ByteValue *bv = EncryptedContent.GetByteValue
        ();

    std::ofstream of( outfile );
    of.write( bv->GetPointer(), bv->GetLength() );
    of.close();

    return 0;
}

```

27.40 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 true;
}

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" );
            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;
}

```

27.41 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 );
    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);
    //gdcmErrorMacro( "opj_decode failed" );
    return 1;
}

    opj_cp_t * cp = ((opj_jp2_t*)dinfo->jp2_handle)->j2k->cp;
    opj_tcp_t *tcp = &cp->tccps[0];
    opj_tccp_t *tccp = &tcp->tccps[0];
    /*      std::cout << "\n No of Cols In Image" << image->x1;
    std::cout << "\n No of Rows In Image" << image->y1;
    std::cout << "\n No of Components in Image" << image->numcomps;
    std::cout << "\n No of Resolutions"<< tccp->numresolutions << "\n";
*/

    opj_j2k_t* j2k = NULL;
    opj_jp2_t* jp2 = NULL;
    jp2 = (opj_jp2_t*)dinfo->jp2_handle;
    int reversible = jp2->j2k->cp->tccps->tccps->qmfbid;
    //std::cout << reversible;
    int compno = 0;
    opj_image_comp_t *comp = &image->comps[compno];
    int Dimensions[2];
    Dimensions[0]= comp->w;
    Dimensions[1] = comp->h;
    opj_cio_close(cio);
    unsigned long len = Dimensions[0]*Dimensions[1] * image->numcomps;
    //std::cout << "\nTest" <<image->comps[0].factor;
    char *raw = new char[len];
    for (unsigned int compno = 0; compno < (unsigned int)image->numcomps; compno
        ++){
        {

```

```

    obj_image_comp_t *comp = &image->comps[compno];

    int w = image->comps[compno].w;
    int h = image->comps[compno].h;
    uint8_t *data8 = (uint8_t*)raw + compno;
    for (int i = 0; i < w * h; i++)
    {
        int v = image->comps[compno].data[i];
        *data8 = (uint8_t)v;
        data8 += image->numcomps;
    }
}

gdcm::Writer w;
gdcm::File &file = w.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

file.GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian
);

gdcm::UIDGenerator uid;
gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP
Instance UID
de.SetVR( gdcm::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );

gdcm::DataElement del( gdcm::Tag(0x8,0x16) );
del.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms( gdcm::MediaStorage::CTImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( del );

const char mystr[] = "MONOCHROME2 ";
gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
de2.SetVR( gdcm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcm::Attribute<0x0028,0x0010> row = {image->
    comps[0].w};
//row.SetValue(512);
ds.Insert( row.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcm::Attribute<0x0028,0x0011> col = {image->
    comps[0].h};
ds.Insert( col.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0008>
    Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0002> at1 = {image->
    numcomps};
ds.Insert( at1.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );

if (flag == 1)
{
    for (int i=0; i < No_Of_Resolutions; i++)
    {
        int a = 1;
        int b = 1;

        while(a!=(No_Of_Resolutions)-i))
        {

```

```

        b = b*2;
        a = a+1;
    }
    uint16_t row = (image->y1)/b;
    uint16_t col = (image->x1)/b;
    //std::cout << row;
    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
);

gdcm::UIDGenerator uid1;
gdcm::DataElement dea( gdcm::Tag(0x8,0x18) ); //
    SOP Instance UID
dea.SetVR( gdcm::VR::UI );
const char *ul = uid1.Generate();
dea.SetByteValue( ul, strlen(ul) );
ds1.Insert( dea );

gdcm::DataElement deb( gdcm::Tag(0x8,0x16) );
deb.SetVR( gdcm::VR::UI );
gdcm::MediaStorage msl(
    gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
deb.SetByteValue( msl.GetString(), strlen( msl.GetString() ) );
ds1.Insert( deb );

const char mystr1[] = "MONOCHROME2 ";
gdcm::DataElement dec( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
dec.SetVR( gdcm::VR::CS );
dec.SetByteValue(mystr, strlen(mystr1));
ds1.Insert( dec );

gdcm::Attribute<0x0028,0x0010> row1 = {image->
    y1};
//row.SetValue(512);
ds1.Insert( row1.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcm::Attribute<0x0028,0x0011> col1 = {image->
    x1};
ds1.Insert( col1.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0008>
    Number_Of_Frames1 = {tccp->numresolutions};
ds1.Insert( Number_Of_Frames1.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0100> ata = {8};

```

```

    ds1.Insert( ata.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0002> atb = {image->
        numcomps};
    ds1.Insert( atb.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0101> atc = {8};
    ds1.Insert( atc.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0102> atd = {7};
    ds1.Insert( atd.GetAsDataElement() );

    theStreamWriter.SetFile(file1);

    gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
    des.SetVR(gdcm::VR::SQ);
    //des.SetVR(gdcm::VM::VM1);
    des.SetValue(*sq);
    des.SetVLToUndefined();

    ds1.Insert(des);

    if (!theStreamWriter.WriteImageInformation()){
        std::cerr << "unable to write image information" << std::endl;
        return 1; //the CanWrite function should prevent getting here, else,
        //that's a test failure
    }
}

theStreamWriter.SetFile(file);

if (!theStreamWriter.CanWriteFile()){
    delete [] raw;
    std::cout << "Not able to write";
    return 0; //this means that the file was unwritable, period.
    //very similar to a ReadImageInformation failure
}
else
    std::cout<<"\nable to read";

// Important to write here
std::vector<unsigned int> extent = gdcm::ImageHelper::GetDimensionsValue
    (file);

    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 4;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of
        theChunkSize
    unsigned short zmax = extent[2];
    std::cout << "\n" << xmax << "\n" << ymax << "\n" << zmax << "\n" << image->
        numcomps << "\n";

    if (xmax == 0 || ymax == 0)
    {
        std::cerr << "Image has no size, unable to write zero-sized image." <<
        std::endl;
        return 0;
    }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure
        to grab
    //the bytes sequentially. So, store how far you got in the buffer with
        each iteration.
    for (z = 0; z < zmax; ++z){
        for (y = 0; y < ymax; y += ychunk){
            nexty = y + ychunk;
            if (nexty > ymax) nexty = ymax;
            theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z
            , z+1);
            unsigned long len = theStreamWriter.DefineProperBufferLength
            ();
            std::cout << "\n" << len;
            char* finalBuffer = new char[len];
            memcpy(finalBuffer, &(raw[prevLen]), len);
            std::cout << "\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 = 0x00dd;
    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, theBufferSizel);
of.flush();
assert( of );

return 0;
}

```

27.42 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 );

gdcmm::DataElement del( gdcmm::Tag(0x8,0x16) );
del.SetVR( gdcmm::VR::UI );
gdcmm::MediaStorage ms(
    gdcmm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( del );

const char mystr[] = "RGB";
gdcmm::DataElement de2( gdcmm::Tag(0x28,0x04) );
//de.SetTag(gdcmm::Tag(0x28,0x04));
de2.SetVR( gdcmm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcmm::Attribute<0x0028,0x0010> row = {256};
//row.SetValue(512);
ds.Insert( row.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcmm::Attribute<0x0028,0x0011> col = {256};
ds.Insert( col.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0008>
    Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0002> at1 = {3}; //
    bits per pixel
ds.Insert( at1.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0006> at4 = {0};
ds.Insert( at4.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0103> at5 = {0};
ds.Insert( at5.GetAsDataElement() );

//de.SetTag(gdcmm::Tag(0x7fe0,0x0010));
//ds.Insert(de);

gdcmm::StreamImageWriter theStreamWriter;
gdcmm::SmartPointer<gdcmm::SequenceOfItems>
    sq = new gdcmm::SequenceOfItems();
sq->SetLengthToUndefined();

uint16_t row1 = 256;
uint16_t col1 = 256;
//std::cout << row;

gdcmm::Element<gdcmm::VR::IS, gdcmm::VM::VM1>
    el2;
el2.SetValue(1);
gdcmm::DataElement rfn = el2.GetAsDataElement()
(); //rfn ---> reference frame number
rfn.SetTag( gdcmm::Tag(0x0008,0x1160) );

gdcmm::Element<gdcmm::VR::US, gdcmm::VM::VM2>
    el;
el.SetValue(1,0);
el.SetValue(1,1);
gdcmm::DataElement ulr = el.GetAsDataElement()
(); //ulr --> upper left col/row
ulr.SetTag( gdcmm::Tag(0x0048,0x0201) );

gdcmm::Element<gdcmm::VR::US, gdcmm::VM::VM2>
    el1;
el1.SetValue(col1,0);
el1.SetValue(row1,1);

```

```

gdcmm::DataElement brr = ell.GetAsDataElement
();
brr.SetTag( gdcmm::Tag(0x0048,0x0202) );          //brr
--> bottom right col/row

gdcmm::Item it;
gdcmm::DataSet &nds = it.GetNestedDataSet();
nds.Insert( rfn );
nds.Insert(ulr);
nds.Insert(brr);

sq->AddItem(it);

gdcmm::DataElement des( gdcmm::Tag(0x0048,0x0200) );
des.SetVR(gdcmm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(des);

theStreamWriter.SetFile(file);

std::ofstream of;
of.open( "output.dcm", std::ios::out | std::ios::binary );
theStreamWriter.SetStream(of);

if (!theStreamWriter.CanWriteFile()){
    delete [] buffer;
    std::cout << "Not able to write";
    return 0; //this means that the file was unwritable, period.
    //very similar to a ReadImageInformation failure
}
else
    std::cout<<"\nable to read";

if (!theStreamWriter.WriteImageInformation()){
    std::cerr << "unable to write image information" << std::endl;
    delete [] buffer;
    return 1; //the CanWrite function should prevent getting here, else,
    //that's a test failure
}

std::vector<unsigned int> extent =
    gdcmm::ImageHelper::GetDimensionsValue
        (file);

unsigned short xmax = extent[0];
unsigned short ymax = extent[1];
unsigned short theChunkSize = 1;
unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of
    theChunkSize
unsigned short zmax = extent[2];

std::cout << xmax << ymax << zmax;

if (xmax == 0 || ymax == 0)
{
    std::cerr << "Image has no size, unable to write zero-sized image." <<
    std::endl;
    return 0;
}

int z, y, nexty;
unsigned long prevLen = 0; //when going through the char buffer, make sure
    to grab
//the bytes sequentially. So, store how far you got in the buffer with
    each iteration.
for (z = 0; z < zmax; ++z){
    for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z
        , z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength
        ();
        std::cout << "\n" << len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(buffer[prevLen]), len);
        std::cout << "\nable to write";
    }
}

```

```

        if (!theStreamWriter.Write(finalBuffer, len)){
            std::cerr << "writing failure:" << "output.dcm" << " at y = " << y <<
            " and z= " << z << std::endl;
            delete [] buffer;
            delete [] finalBuffer;
            return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
    }
    delete buffer;

    uint16_t firstTag1 = 0xfffe;
    uint16_t secondTag1 = 0xe0dd;
    uint32_t thirdTag1 = 0x00000000;
    //uint16_t fourthTag1 = 0xffff;
    const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
    char* tmpBuffer2 = new char[theBufferSize1];
    memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
    memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
    memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
    //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
    assert( of && !of.eof() && of.good() );
    of.write(tmpBuffer2, theBufferSize1);
    of.flush();
    assert( of );

    return 0;
}

```

27.43 FindAllPatientName.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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 can use the gdcm.CompositeNetworkFunctions class
for executing a C-FIND query
It will print the list of patient name found

Usage:

python FindAllPatientName.py

"""

import gdcm

# Patient Name
tag = gdcm.Tag(0x10,0x10)
de = gdcm.DataElement(tag)

# Search all patient name where string match 'F*'
de.SetByteValue('F*',gdcm.VL(2))

ds = gdcm.DataSet()
ds.Insert(de)

cnf = gdcm.CompositeNetworkFunctions()
theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)

#print theQuery.ValidateQuery()

# prepare the variable for output
ret = gdcm.DataSetArrayType()

```

```
# Execute the C-FIND query
cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')

for i in range(0,ret.size()):
    print "Patient #",i
    print ret[i]
```

27.44 FixBrokenJ2K.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 "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".
 */
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 &pixeldata = file.GetDataSet
    ().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
    const gdcm::SequenceOfFragments *sqf = pixeldata.
    GetSequenceOfFragments();
    if( !sqf )
    {
        return 1;
    }
    const gdcm::Fragment &frag = sqf->GetFragment(0);

    const gdcm::ByteValue *bv = frag.GetByteValue();
    const char *ptr = bv->GetPointer();
    size_t len = bv->GetLength();
```

```

const char sig[] = "\x00\x00\x00\x00\x6A\x70\x32\x63";
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(0x7fe0,0
            x0010) );
        pixeldata.SetVR( gdcm::VR::OB );
        gdcm::SmartPointer<gdcm::SequenceOfFragments>
            sq = new gdcm::SequenceOfFragments;

        gdcm::Fragment frag;
        // remove 8 first bytes:
        frag.SetByteValue( ptr + 8, 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;
}

```

27.45 FixCommaBug.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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 LC_NUMERIC set to something not compatible with "C" it is possible to
write out "," instead of
"." as required by the DICOM standard
Issue is still current (IMHO) with gdcm 2.0.9
"""

import gdcm
import sys

filename = sys.argv[1]
outname = sys.argv[2]

# read
r = gdcm.Reader()
r.SetFileName( filename )
if not r.Read():
    print "not valid"
    sys.exit(1)

file = r.GetFile()
dataset = file.GetDataSet()

ano = gdcm.Anonymizer()
ano.SetFile( file )

tags = [
    gdcm.Tag(0x0018,0x1164),
    gdcm.Tag(0x0018,0x0088),
    gdcm.Tag(0x0018,0x0050),
    gdcm.Tag(0x0028,0x0030),
]

for tag in tags:
    print tag
    if dataset.FindElement( tag ):
        pixelspacing = dataset.GetDataElement( tag )
        #print pixelspacing
        bv = pixelspacing.GetByteValue()
        str = bv.GetBuffer()
        #print bv.GetLength()
        #print len(str)
        new_str = str.replace(",",".")
        # Need to explicitly pass bv.GetLength() to remove any trailing garbage
        ano.Replace( tag, new_str, bv.GetLength() )

#print dataset

w = gdcm.Writer()
w.SetFile( file )
w.SetFileName( outname )
if not w.Write():
    print "Cannot write"
    sys.exit(1)

# paranoid:
image_reader = gdcm.ImageReader()
image_reader.SetFileName( outname )
if not image_reader.Read():
    print "there is still a comma"
    sys.exit(1)

print "Sucess!"
sys.exit(0) # success

```

27.46 FixJAIBugJPEGLS.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

```

All rights reserved.
See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"

#include <fstream>

#include "gdcm_charls.h"

/*
 * This small example should show how one can handle the famous JAI-JPEGLS bug
 * It will take in as invalid DICOM/JAI-JPEG-LS and write out as Explicit
 * Little
 * Endian. One can use 'gdcmconv --jpegl's' to recompress properly
 *
 * References:
 * http://charls.codeplex.com/discussions/230307?ProjectName=charls
 * http://charls.codeplex.com/workitem/7297
 * http://www.dcm4che.org/jira/browse/DCM-442
 * http://www.dcm4che.org/jira/browse/DCMEE-1144
 * http://java.net/jira/browse/JAI_IMAGEIO_CORE-183
 *
 * Explanation of the issue:
 *
 * Seems, the error is in the calculation of the default values for thresholds
 * T1,
 * T2, T3, in particular min(MAXVAL, 4095) is not applied in
 *
 * FACTOR = (min(MAXVAL, 4095) + 128)/256
 *
 * as specified in http://www.itu.int/rec/T-REC-T.87-199806-I/en .
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::FileMetaInformation::SetSourceApplicationEntityTitle
        ( "FixJAIBugJPEGLS" );

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::Image &image = reader.GetImage();
    //unsigned long len = image.GetBufferLength();
    const gdcm::DataElement &in =
        reader.GetFile().GetDataSet().GetDataElement
            ( gdcm::Tag(0x7fe0,0x0010) );
    const gdcm::SequenceOfFragments *sf = in.
        GetSequenceOfFragments();
    if( !sf )
    {
        std::cerr << "No pixel data (or not encapsulated)" << std::endl;
        return 1;
    }
    const unsigned int *dims = image.GetDimensions();
    if ( sf->GetNumberOfFragments() != dims[2] )
    {
        std::cerr << "Unsupported" << std::endl;
        return 1;
    }

    // unsigned long totalLen = sf->ComputeByteLength();
    std::vector<BYTE> rgbyteOutall;
    for(unsigned int i = 0; i < sf->GetNumberOfFragments(); +

```

```

        +i)
    {
        const gdcm::Fragment &frag = sf->GetFragment(i);
        if( frag.IsEmpty() ) return 1;
        const gdcm::ByteValue *bv = frag.GetByteValue();
        if( !bv ) return 1;
        unsigned long totalLen = bv->GetLength();

        std::vector<char> vbuffer;
        vbuffer.resize( totalLen );
        char *buffer = &vbuffer[0];
        bv->GetBuffer(buffer, totalLen);
        const BYTE* pbyteCompressed0 = (const BYTE*)buffer;
        while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
        {
            totalLen--;
        }

#ifdef GDCM_USE_SYSTEM_CHARLS
        JlsParameters metadata;
#else
        JlsParamaters metadata;
#endif
        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" );
        of.write( &vbuffer[0], vbuffer.size() );
        of.close();
    #endif

    const char *pbyteCompressed = &vbuffer[0];
    unsigned int cbyteCompressed = vbuffer.size(); // updated legnth

    #ifdef GDCM_USE_SYSTEM_CHARLS
        JlsParameters params = {0};
    #else
        JlsParamaters params = {0};
    #endif
    JpegLsReadHeader( pbyteCompressed, cbyteCompressed, &params);

    std::vector<BYTE> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize( params.height * params.width * ((params.bitspersample + 7)
        / 8) * params.components);

    #ifdef GDCM_USE_SYSTEM_CHARLS
        JLS_ERROR result =
            JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed,
                cbyteCompressed, &params );
    #else
        JLS_ERROR result =
            JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed,
                cbyteCompressed );
    #endif
    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(0x7fe0,0
        x0010) );
    pixeldata.SetVR( gdcm::VR::OW );
    pixeldata.SetByteValue( (char*)&rgbyteOutall[0], 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;
}

```

27.47 gdcmmorthoplanes.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 "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 "gdcmmSystem.h"
#include "gdcmmDirectory.h"
#include "gdcmmIPPSorter.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 && gdcm::System::FileIsDirectory
            ( filename ) )
        {
            std::cout << "Loading directory: " << filename << std::endl;
            bool recursive = false;
            gdcm::Directory d;
            d.Load(filename, recursive);
            gdcm::Directory::FileNamesType const &files
                = d.GetFilesNames();
            for( gdcm::Directory::FileNamesType::const_iterator it = files.begin();
                it != files.end(); ++it )
            {
                filenames.push_back( it->c_str() );
            }
        }
        else // list of files passed directly on the cmd line:
            // discard non-existing or directory
        {
            for(int i=1; i < argc; ++i)
            {
                filename = argv[i];
                if( gdcm::System::FileExists( filename ) )
                {
                    if( gdcm::System::FileIsDirectory(
                        filename ) )
                    {
                        std::cerr << "Discarding directory: " << filename << std::endl;
                    }
                    else
                    {
                        filenames.push_back( filename );
                    }
                }
            }
        }
    }
}

```

```

        else
        {
            std::cerr << "Discarding non existing file: " << filename <<
            std::endl;
        }
    }
}
//names->Print( std::cout );
}

//gdcm::Trace::DebugOn();
//gdcm::Trace::WarningOn();
gdcm::IPPSorter s;
s.SetComputeZSpacing( true );
s.SetZSpacingTolerance( 1e-3 );
bool b = s.Sort( filenames );
if( !b )
{
    std::cerr << "Failed to sort files" << std::endl;
    return 1;
}
std::cout << "Sorting succeeded:" << std::endl;
s.Print( std::cout );

std::cout << "Found z-spacing:" << std::endl;
std::cout << s.GetZSpacing() << std::endl;
double 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() );
}

//delete[] fname;
vtkGDCMImageReader * reader = vtkGDCMImageReader::New
();
//reader->SetFileLowerLeft( 1 );
reader->SetFileNames( files );
reader->Update(); // important
//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();
v16->SetInput( reader->GetOutput() );
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

files->Delete();

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();
planeWidgetX->SetInput(v16->GetOutput());
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();
planeWidgetY->SetInput(v16->GetOutput());
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();
planeWidgetZ->SetInput(v16->GetOutput());
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();
colorMap->SetInput(planeWidgetZ->GetResliceOutput());
colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

vtkImageActor* imageActor = vtkImageActor::New();
imageActor->PickableOff();
imageActor->SetInput(colorMap->GetOutput());

// 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;
}

```

27.48 gdcmmreslice.cxx

```

/*=====

```

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkImageFlip.h"
#include "vtkImageReslice.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"

int main( int argc, char *argv[] )
{
    if( argc < 2 ) return 1;
    vtkGDCMImageReader *reader = vtkGDCMImageReader::New
        ();
    reader->SetFileName( argv[1] );
    //reader->FileLowerLeftOn();
    reader->Update();

    vtkImageFlip *flip = vtkImageFlip::New();
    flip->SetInput( reader->GetOutput() );
    flip->SetFilteredAxis(0);
    flip->Update();

    vtkImageReslice *reslice = vtkImageReslice::New();
    //reslice->SetInput( reader->GetOutput() );
    reslice->SetInput( flip->GetOutput() );
    //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();
    texture->SetInput(ima);
    texture->InterpolateOn();
    texture->SetLookupTable(table);

    // PlaneSource
    vtkPlaneSource* plane = vtkPlaneSource::New();

    // PolyDataMapper
    vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();

```

```

planeMapper->SetInput(plane->GetOutput());

// 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;
}

```

27.49 gdcmrtnonplan.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```

```

=====*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkXMLPolyDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>

#include "gdcmmReader.h"
#include "gdcmmAttribute.h"

/*
  This example is just for fun. We found a RT Ion Plan Storage and simply
  extracted the viz stuff for VTK

  RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " filename.dcm outfile.vti\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];
    const char * outfilename2 = argv[3];

    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcmm::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( ms != gdcmm::MediaStorage::RTIonPlanStorage
        )
    {
        return 1;
    }

    /*
    (300a,03a2) SQ # u/1,1 Ion
        Beam Sequence
    (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/1,1 Ion
    Range Compensator Sequence
(fffe,e000) na (Item with undefined length)
    (300a,00e1) SH [lucite] # 6,1
    Material ID
    (300a,00e4) IS [1 ] # 2,1
    Compensator Number
    (300a,00e5) SH [75hdhe5 ] # 8,1
    Compensator ID
    (300a,00e7) IS [35] # 2,1
    Compensator Rows
    (300a,00e8) IS [37] # 2,1
    Compensator Columns
    (300a,00e9) DS [3.679991\4.249288 ] # 18,2
    Compensator Pixel Spacing
    (300a,00ea) DS [-76.00\62.50] # 12,2
    Compensator Position
    (300a,00ec) DS
    [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38
    .79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.8
    8\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\43.52\52.13\52.13\52.13\53.18\53.52\54.0]
    Thickness Data
    (300a,02e0) CS [ABSENT] # 6,1
    Compensator Divergence
    (300a,02e1) CS [SOURCE_SIDE ] # 12,1
    Compensator Mounting Position
    (300a,02e4) FL 39.2 # 4,1
    Isocenter to Compensator Tray Distance
    (300a,02e5) FL 2.12 # 4,1
    Compensator Column Offset
    (300a,02e8) FL 4.76 # 4,1
    Compensator Milling Tool Diameter
(fffe,e00d)
*/
const gdcmm::DataSet& ds = reader.GetFile().GetDataSet
();
gdcmm::Tag tbeamsq(0x300a,0x03a2);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcmm::DataElement &tbeamsq = ds.GetDataElement
( tbeamsq );
//std::cout << tbeamsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems>
sqi = tbeamsq.GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//const gdcmm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcmm::Item & item = sqi->GetItem(1); // Item start
    at #1
const gdcmm::DataSet& nestedds = item.GetNestedDataSet
();
//std::cout << nestedds << std::endl;
gdcmm::Tag tcompensatorsq(0x300a,0x02ea);
if( !nestedds.FindDataElement( tcompensatorsq ) )
{
    return 1;
}
const gdcmm::DataElement &tcompensatorsq = nestedds.
GetDataElement( tcompensatorsq );
//std::cout << tcompensatorsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems>
ssqi = tcompensatorsq.GetValueAsSQ();
const gdcmm::Item & item2 = ssqi->GetItem(1); // Item start
    at #1
const gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet
();
//std::cout << nestedds2 << std::endl;
gdcmm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
if( !nestedds2.FindDataElement( tcompensatorthicknessdata )
)
{
    return 1;
}

```

```

const gdcmm::DataElement &compensatorthicknessdata =
    nestedds2.GetDataElement( tcompensatorthicknessdata );
// std::cout << compensatorthicknessdata << std::endl;
gdcmm::Attribute<0x300a,0x00ec> at;
at.SetFromDataElement( compensatorthicknessdata );
const double* pts = at.GetValues();
// (300a,00e7) IS [35] # 2,1
    Compensator Rows
gdcmm::Attribute<0x300a,0x00e7> at1;
const gdcmm::DataElement &compensatorrows = nestedds2.
    GetDataElement( at1.GetTag() );
at1.SetFromDataElement( compensatorrows );
std::cout << at1.GetValue() << std::endl;
// (300a,00e8) IS [37] # 2,1
    Compensator Columns
gdcmm::Attribute<0x300a,0x00e8> at2;
const gdcmm::DataElement &compensatorcols = nestedds2.
    GetDataElement( at2.GetTag() );
at2.SetFromDataElement( compensatorcols );
std::cout << at2.GetValue() << std::endl;

    // (300a,00e9) DS [3.679991\4.249288 ] # 18,2
    Compensator Pixel Spacing
gdcmm::Attribute<0x300a,0x00e9> at3;
const gdcmm::DataElement &compensatorpixelspacing =
    nestedds2.GetDataElement( at3.GetTag() );
at3.SetFromDataElement( compensatorpixelspacing );
std::cout << at3.GetValue(0) << std::endl;
// (300a,00ea) DS [-76.00\62.50] # 12,2
    Compensator Position
gdcmm::Attribute<0x300a,0x00ea> at4;
const gdcmm::DataElement &compensatorposition = nestedds2.
    GetDataElement( at4.GetTag() );
at4.SetFromDataElement( compensatorposition );
std::cout << at4.GetValue(0) << std::endl;

vtkDoubleArray *d = vtkDoubleArray::New();
d->SetArray( (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);
img->SetScalarTypeToDouble();
img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); //
    FIXME image is upside down
img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
img->SetNumberOfScalarComponents(1);
img->GetPointData()->SetScalars(d);

img->Update();
img->Print(std::cout);

vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
writeb->SetInput( img );
writeb->SetFileName( outfilename );
writeb->Write( );
/*
(300a,03a6) SQ # u/1,1 Ion Block
Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [brass ] # 6,1 Material ID
(300a,00f7) FL 95.03 # 4,1 Isocenter to
Block Tray Distance
(300a,00f8) CS [APERTURE] # 8,1 Block Type
(300a,00fa) CS [ABSENT] # 6,1 Block
Divergence
(300a,00fb) CS [SOURCE_SIDE ] # 12,1 Block Mounting
Position
(300a,00fc) IS [1 ] # 2,1 Block Number
(300a,0100) DS [50.00 ] # 6,1 Block Thickness
(300a,0104) IS [179 ] # 4,1 Block Number of
Points
(300a,0106) DS
[1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47
.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2\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\40.0\39.5\41.5\37.9\42
2\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
(fffe,e00d)
(fffe,e0dd)

```

```

*/
gdcmm::Tag tblocksq(0x300a,0x03a6);
if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcmm::DataElement &tblocksq = nestedds.GetDataElement
( tblocksq );
//std::cout << tblocksq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems>
    sssqi = tblocksq.GetValueAsSQ();
const gdcmm::Item &item3 = sssqi->GetItem(1); // Item
    start at #1
const gdcmm::DataSet& nestedds3 = item3.GetNestedDataSet
();

gdcmm::Tag tblockdata(0x300a,0x0106);
if( !nestedds3.FindDataElement( tblockdata ) )
{
    return 1;
}
const gdcmm::DataElement &tblockdata = nestedds3.
    GetDataElement( tblockdata );
// std::cout << tblockdata << std::endl;
gdcmm::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( tblockdata );

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 &tbnpts = nestedds3.
    GetDataElement( bnpts.GetTag() );
bnpts.SetFromDataElement( tbnpts );
//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] = ptr[2*i+0];
    x[1] = ptr[2*i+1];
    //x[2] = ptr[2*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();
output->Update();
output->Print( std::cout );

// }

vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

vtkImageColorViewer *viewer = vtkImageColorViewer::New
();

```

```

viewer->SetInput(img);
viewer->SetupInteractor(iren);
viewer->SetSize(600, 600);
viewer->GetRenderer()->ResetCameraClippingRange();
viewer->Render();
viewer->GetRenderer()->ResetCameraClippingRange();

vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
//vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
    cubeMapper->SetInput( output );
    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();
    writec->SetInput( output );
    writec->SetFileName( outfilename2 );
    writec->Write( );

    iren->Initialize();
    iren->Start();

    return 0;
}

```

27.50 gdcmrtpplan.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>

#include "gdcmReader.h"
#include "gdcmAttribute.h"

/*
This example is just for fun. We found a fake RT Ion Plan Storage and simply
extracted the viz stuff for VTK
but this is rather a RT Plan storage
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " filename.dcm outfile.vti\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];

```

```

gdcmm::Reader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    return 1;
}

gdcmm::MediaStorage ms;
ms.SetFromFile( reader.GetFile() );
if( ms != gdcmm::MediaStorage::RTIonPlanStorage
    )
{
    return 1;
}

/*
(300a,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\33.28\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.8
        8\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\43.52\52.13\52.13\52.13\53.18\53.52\54.0]
        Thickness 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.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcmm::DataElement &tbeamsq = ds.GetDataElement
    ( tbeamsq );
//std::cout << tbeamsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems>
    sqi = tbeamsq.GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

```

```

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcm::Item & item = sqi->GetItem(2); // Item start
  at #1
const gdcm::DataSet& nestedds = item.GetNestedDataSet
  ();
//std::cout << nestedds << std::endl;
gdcm::Tag tcompensatorsq(0x300a,0x00e3);
if( !nestedds.FindDataElement( tcompensatorsq ) )
{
  return 1;
}
const gdcm::DataElement &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 );
//img->SetExtent(1, xdim, 1, ydim, 1, zdim);
img->SetScalarTypeToDouble();
img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); //
  FIXME image is upside down
img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
img->SetNumberOfScalarComponents(1);
img->GetPointData()->SetScalars(d);

```

```

vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
writeb->SetInput( img );
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\40.0\39.5\41.5\37.9\42.2\37.4\43.0\37.1\44.7\36]
    (fffe,e00d)
  (fffe,e000) na (Item with undefined length)
  (fffe,e00d)
(fffe,e0dd)
*/

gdcmm::Tag tblocksq(0x300a,0x00f4);
if( !nestedds.FindDataElement( tblocksq ) )
{
  return 1;
}
const gdcmm::DataElement &blocksq = nestedds.GetDataElement
( tblocksq );
//std::cout << blocksq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems>
  sssqi = blocksq.GetValueAsSQ();
const gdcmm::Item & item3 = sssqi->GetItem(1); // Item
  start at #1
const gdcmm::DataSet& nestedds3 = item3.GetNestedDataSet
();

gdcmm::Tag tblockdata(0x300a,0x0106);
if( !nestedds3.FindDataElement( tblockdata ) )
{
  return 1;
}
const gdcmm::DataElement &blockdata = nestedds3.
  GetDataElement( tblockdata );
// std::cout << blockdata << std::endl;
gdcmm::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( blockdata );

vtkDoubleArray *scalars = vtkDoubleArray::New();
scalars->SetNumberOfComponents(3);

gdcmm::Attribute<0x300a,0x0104> bnpts; // IS
  [179 ] # 4,1 Block Number of Points
if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
{
  return 1;
}
const gdcmm::DataElement &blocknpts = nestedds3.
  GetDataElement( bnpts.GetTag() );
bnpts.SetFromDataElement( blocknpts );
std::cout << bnpts.GetValue() << std::endl;

vtkPolyData *output = vtkPolyData::New();
vtkPoints *newPts = vtkPoints::New();
vtkCellArray *polys = vtkCellArray::New();
const double *ptr = at_.GetValues();
//unsigned int npts = bnpts.GetNumberOfValues() / 2;
unsigned int npts = bnpts.GetValue();
vtkIdType *ptIds = new vtkIdType[npts];
for(unsigned int i = 0; i < npts; ++i)
{
  float x[3] = {};

```

```

        x[0] = ptr[2*i+0];
        x[1] = ptr[2*i+1];
        //x[2] = pts[i+2];
        vtkIdType ptId = newPts->InsertNextPoint( x );
        //std::cout << x[0] << ", " << x[1] << ", " << x[2] << std::endl;
        ptIds[i] = ptId;
    }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
    output->Update();
    output->Print( std::cout );

// }

    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

    vtkImageColorViewer *viewer = vtkImageColorViewer::New
    ();
    viewer->SetInput(img);
    viewer->SetupInteractor(iren);
    viewer->SetSize(600, 600);
    viewer->Render();

    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
    cubeMapper->SetInput( output );
    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;
}

```

27.51 gdcmscene.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMPolyDataReader.h"
//#include "vtkGDCMPolyDataWriter.h"

#include "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)
    {
        append->AddInput( reader->GetOutput(i) );
    }

    vtkPolyDataWriter * writer = vtkPolyDataWriter::New();
    writer->SetInput( reader->GetOutput() );
    writer->SetFileName( "rtstruct.vtk" );
    //writer->Write();

    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
    //cubeMapper->SetInput( reader->GetOutput() );
    cubeMapper->SetInput( append->GetOutput() );
    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;
}

```

27.52 gdcmttexture.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 "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[] )
{
    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();
    texture->SetInput(ima);
    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();
planeMapper->SetInput(plane->GetOutput());

// 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;
}

```

27.53 gdcmvolume.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "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[])
{
    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;
}

```

27.54 GenAllVR.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 "gdcmlGlobal.h"
#include "gdcmlDummyValueGenerator.h"
#include "gdcmlMediaStorage.h"
#include "gdcmlWriter.h"
#include "gdcmlItem.h"
#include "gdcmlImageReader.h"
#include "gdcmlSequenceOfItems.h"
#include "gdcmlFile.h"
#include "gdcmlTag.h"
#include "gdcmlDict.h"
#include "gdcmlDictEntry.h"
#include "gdcmlDicts.h"
#include "gdcmlTransferSyntax.h"
#include "gdcmlUIDGenerator.h"
#include "gdcmlFileExplicitFilter.h"

#include <cstdlib>
#include <cstring>

gdcml::Tag FindTagFromVR(gdcml::Dict const &dict, gdcml::VR
    const &vr)
{
    using gdcml::Dict;
    Dict::ConstIterator beg = dict.Begin();
    Dict::ConstIterator end = dict.End();
    Dict::ConstIterator it;
    for( it = beg; it != end; ++it)
    {
        const gdcml::Tag &t = it->first;
        const gdcml::DictEntry &de = it->second;
        const gdcml::VR &vr_de = de.GetVR();
        if( vr == vr_de && !de.GetRetired() && t.GetGroup() >=
            0x8 )
        {
            return t;
        }
    }
    return gdcml::Tag(0xffff,0xffff);
}

struct rnd_gen {
    rnd_gen(char const* range = "abcdefghijklmnopqrstuvwxyz0123456789")
        : range(range), len(std::strlen(range)) { }
}

```

```

char operator ()() const {
    return range[static_cast<std::size_t>(std::rand() * (1.0 / (RAND_MAX + 1.0
    )) * len)];
}
private:
    char const* range;
    std::size_t len;
};

/*
 */
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    const char *outfilename = argv[1];
    static const gdcm::Global &g = gdcm::Global::GetInstance
        ();
    static const gdcm::Dicts &dicts = g.GetDicts();
    static const gdcm::Dict &pubdict = dicts.GetPublicDict
        ();
    using gdcm::VR;
    using gdcm::Tag;

    gdcm::Writer w;

    gdcm::File &f = w.GetFile();
    gdcm::DataSet &ds = f.GetDataSet();

    gdcm::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( true );
    fef.SetFile( w.GetFile() );
    if( !fef.Change() )
    {
        std::cerr << "Failed to change" << std::endl;
        return 1;
    }

    gdcm::SmartPointer<gdcm::SequenceOfItems>
        sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();

    // gdcm::DummyValueGenerator dv;

    const std::size_t len = 10;
    char ss[len+1];
    ss[len] = '\0';

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
    owner.SetByteValue(owner_str, strlen(owner_str));
    owner.SetVR( gdcm::VR::LO );

    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    // nds.Insert(owner);
    // nds.Insert(de);

    // Insert sequence into data set
    gdcm::DataElement des( gdcm::Tag(0x4d4d, 0x1001) );
    des.SetVR(gdcm::VR::SQ);
    des.SetValue(*sq);
    des.SetVLToUndefined();

    ds.Insert(owner);
    ds.Insert(des);

    // avoid INVALID = 0
    for(int i = 1; i < 27; ++i)
    {
        VR vr = (VR::VRType)(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, 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, 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(), 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;
}

```

27.55 GenerateDICOMDIR.cs

This is a C# example on how to use gdcmm::DICOMDIRGenerator

```

/*=====
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.
=====*/

/*
 * Simple C# example to show how to use DICOMDIRGenerator
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcmm/debug-gcc/bin
 * $ mono bin/GenerateDICOMDIR.exe path output_filename
 */
using System;
using gdcmm;

public class GenerateDICOMDIR
{
    public static int Main(string[] args)
    {
        string directory = args[0];
        string outfilename = args[1];
    }
}

```

```

Directory d = new Directory();
uint nfiles = d.Load( directory, true );
if(nfiles == 0) return 1;
//System.Console.WriteLine( "Files:\n" + d.toString() );

// Implement fast path ?
// Scanner s = new Scanner();

string descriptor = "My_Descriptor";
FileNamesType filenames = d.GetFilesNames();

gdcm.DICOMDIRGenerator gen = new DICOMDIRGenerator();
gen.SetFilenames( filenames );
gen.SetDescriptor( descriptor );
if( !gen.Generate() )
{
    return 1;
}

gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "GenerateDICOMDIR
" );
gdcm.Writer writer = new Writer();
writer.SetFile( gen.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}
}

```

27.56 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 "gdcmDirectoryHelper.h"

using namespace gdcm;

/*
 * 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);
    if (theRTSeries.empty())
    {
        std::cerr << "No RTStructs found for the test, ending." << std::endl;
        return 1;
    }

    Directory::FileNamesType theRTNames =
        DirectoryHelper::GetFileNamesFromSeriesUIDs
            (theDirName, theRTSeries[0]);

    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 );
    writer->SetFileName( std::string(theDirName + "/" + "GDCMTestRTStruct." +
        theRTSeries[0] + ".dcm").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();
    for (int i = 0; i < reader->GetNumberOfOutputPorts(); ++i)
    {
        writer->SetInput(i, reader->GetOutput(i));
        append->AddInput(reader->GetOutput(i));
        std::string theString = reader->GetRTStructSetProperties()->
            GetStructureSetROIName(i);
        roiNames->InsertValue(i, theString);
        theString = reader->GetRTStructSetProperties()->
            GetStructureSetROIGenerationAlgorithm(i);
        roiAlgorithms->InsertValue(i, theString);
        theString = reader->GetRTStructSetProperties()->
            GetStructureSetRTROIInterpretedType(i);
        roiTypes->InsertValue(i, theString);
    }
    //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
    vtkPolyData* blank = vtkPolyData::New();
    writer->SetInput(numMasks-1, blank);
    roiNames->InsertValue(numMasks-1, "blank");
    roiAlgorithms->InsertValue(numMasks-1, "blank");
    roiTypes->InsertValue(numMasks-1, "ORGAN");

    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 );

// Now we'll look at it.
vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
cubeMapper->SetInput( append->GetOutput() );
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();
roiNames->Delete();
roiTypes->Delete();
theProperties->Delete();
roiAlgorithms->Delete();
blank->Delete();

writer->Delete();

return 0;
}

```

27.57 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 argc, char *argv[])
{

    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;
}

```

27.58 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, strlen(str) );
        else
            de.SetByteValue( safe, strlen(safe) );
    }
    else
    {
        // Create an item
        gdcm::Item it;
        it.SetVLToUndefined();
        gdcm::DataSet &nds = it.GetNestedDataSet();
        // Insert sequence into data set
        assert(de.GetVR() == gdcm::VR::SQ );
        gdcm::SmartPointer<gdcm::SequenceOfItems>
            sq = new gdcm::SequenceOfItems();
        sq->SetLengthToUndefined();
        de.SetValue(*sq);
        de.SetVLToUndefined();
        //ds.Insert(de);
    }
}

```

```

    if( !toremove )
    {
        nds.Insert( CreateFakeElement( balcptags[count], true ) );
        countglobal++;
    }
    else
    {
        gdcm::Attribute<0x0008,0x0000> at1 = { 0 };
        // This element has no reason to be 'anonymized'...
        nds.Insert( at1.GetAsDataElement() );
        gdcm::Attribute<0x000a,0x0000> at2 = { 0 };
        nds.Insert( at2.GetAsDataElement() );
    }
    sq->AddItem(it);
}
return de;
}

/*
*/
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    using gdcm::Tag;
    using gdcm::VR;
    const char *outfilename = argv[1];

    std::vector<gdcm::Tag> balcptags =
        gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
            ();

    gdcm::Writer w;
    gdcm::File &f = w.GetFile();
    gdcm::DataSet &ds = f.GetDataSet();

    // Add attribute that need to be anonymized:
    std::vector<gdcm::Tag>::const_iterator it = balcptags.begin();
    for(; it != balcptags.end(); ++it)
    {
        ds.Insert( CreateFakeElement( *it, true ) );
    }

    // Add attribute that do NOT need to be anonymized:
    static const gdcm::Global &g = gdcm::Global::GetInstance
        ();
    static const gdcm::Dicts &dicts = g.GetDicts();
    static const gdcm::Dict &pubdict = dicts.GetPublicDict
        ();

    using gdcm::Dict;
    Dict::ConstIterator dictit = pubdict.Begin();
    for(; dictit != pubdict.End(); ++dictit)
    {
        const gdcm::Tag &dicttag = dictit->first;
        if( dicttag == Tag(0x6e65,0x6146) ) break;
        //const gdcm::DictEntry &dictentry = dictit->second;
        ds.Insert( CreateFakeElement( dicttag, false ) );
    }
    ds.Remove( gdcm::Tag(0x400,0x500) );
    ds.Remove( gdcm::Tag(0x12,0x62) );
    ds.Remove( gdcm::Tag(0x12,0x63) );

    // Make sure to override any UID stuff
    gdcm::UIDGenerator uid;
    gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
    de.SetVR( VR::UI );
    const char *u = uid.Generate();
    de.SetByteValue( u, strlen(u) );
    //ds.Insert( de );
    ds.Replace( de );

    de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
    de.SetVR( VR::UI );
    gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage
        );
    de.SetByteValue( ms.GetString(), strlen(ms.GetString()) );

```

```

ds.Replace( de ); // replace !

gdcm::FileMetaInformation &fmi = f.GetHeader
();
//fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian
);
fmi.SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian
);

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

27.59 GenFakelImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmFileDerivation.h"
#include "gdcmUIDGenerator.h"
//#include "gdcmImageChangePhotometricInterpretation.h"

/*
 * This example shows two things:
 * 1. How to create an image ex-nihilo
 * 2. How to use the gdcm.FileDerivation filter. This filter is meant to create
    "DERIVED" image
 * object. FileDerivation has a simple API where you can reference *all* the
    input image that have been
 * used to generate the image. The API also allows user to specify the purpose
    of reference (see CID 7202,
 * PS 3.16 - 2008), and the image derivation type (CID 7203, PS 3.16 - 2008).
 */
int main(int, char *[])
{
    // Step 1: Fake Image
    gdcm::SmartPointer<gdcm::Image> im = new
        gdcm::Image;

    char * buffer = new char[ 256 * 256 * 3];
    char * p = buffer;
    int b = 128;
    int ybr[3];
    int ybr2[3];
    int rgb[3];

    for(int r = 0; r < 256; ++r)
        for(int g = 0; g < 256; ++g)
            //for(int b = 0; b < 256; ++b)
            {
                rgb[0] = r;
                rgb[1] = g;
                rgb[1] = 128;
                rgb[2] = b;
                ybr[0] = r;
                ybr[1] = g;
                ybr[1] = 128;
                ybr[2] = b;
            }
}

```

```

    ybr2[0] = r;
    ybr2[1] = g;
    ybr2[1] = 128;
    ybr2[2] = b;
    //gdcm::ImageChangePhotometricInterpretation::YBR2RGB(rgb, ybr);
    //gdcm::ImageChangePhotometricInterpretation::RGB2YBR(ybr2, rgb);
    *p++ = ybr2[0];
    *p++ = ybr2[1];
    *p++ = 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(0x7fe0,0
    x0010) );
pixeldata.SetByteValue( buffer, 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( 1130
    72 );
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;
}

```

27.60 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, strlen( owner_str ) );
    owner.SetVR( gdcm::VR::LO );

    size_t nitems = 1000;
    nitems += std::numeric_limits<uint32_t>::max();
    for( unsigned int idx = 0; idx < nitems; ++idx )
    {
        // Create a dataelement
        //gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
        //de.SetByteValue( ptr, ptr_len );
        //de.SetVR( gdcm::VR::OB );

```

```

    // Create an item
    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;
}

```

27.61 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];
gdcm::Reader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    return 1;
}

gdcm::File &file = reader.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

//const unsigned int nitems = 1000;
const unsigned int ptr_len = 42; /*94967296 / nitems; */
//assert( ptr_len == 42949672 );
char *ptr = new char[ptr_len];
memset(ptr,0,ptr_len);

// Create a Sequence
gdcm::SmartPointer<gdcm::SequenceOfItems>
    sq = new gdcm::SequenceOfItems();
sq->SetLengthToUndefined();

const char owner_str[] = "GDCM CONFORMANCE TESTS";
gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
owner.SetByteValue(owner_str, strlen(owner_str));
owner.SetVR( gdcm::VR::LO );

for(unsigned int idx = 0; idx < 10/* nitems*/; ++idx)
{
    // Create a dataelement
    gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
    ;
    de.SetByteValue(ptr, ptr_len);
    de.SetVR( gdcm::VR::OB );

    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert(owner);
    nds.Insert(de);

    sq->AddItem(it);
}

// Insert sequence into data set
gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
des.SetVR(gdcm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(owner);
ds.Insert(des);

gdcm::Writer w;
w.SetFile( file );
//w.SetCheckFileMetaInformation( true );
w.SetFileName( outfile );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

27.62 GetArray.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/GetArray.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class GetArray
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        ImageReader reader = new ImageReader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        Image image = reader.GetImage();

        PixelFormat pixeltype = image.GetPixelFormat();

        if( image.GetNumberOfDimensions() != 2 )
        {
            // For the purpose of the test, exit early on
            return 1;
        }
        uint dimx = image.GetDimension(0);
        uint dimy = image.GetDimension(1);
        uint npixels = dimx * dimy;
        //LookupTable lut = image.GetLUT();
        //uint r1 = lut.GetLUTLength( LookupTable.LookupTableType.RED );
        //byte[] rbuf = new byte[ r1 ];
        //uint r12 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
        //assert r1 == r12;

        //byte[] str1 = new byte[ image.GetBufferLength()];
        //image.GetBuffer( str1 );
        if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT8 )
        {
            System.Console.WriteLine( "Processing UINT8 image type" );
            byte[] str1 = new byte[ npixels ];
            image.GetArray( str1 );
        }
        else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.INT16 )
        {
            System.Console.WriteLine( "Processing INT16 image type" );
            short[] str1 = new short[ npixels ];
            image.GetArray( str1 );
        }
        else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT16 )
        {
            System.Console.WriteLine( "Processing UINT16 image type" );
            ushort[] str1 = new ushort[ npixels ];
            image.GetArray( str1 );
        }
        else
        {
            //System.Console.WriteLine( "Default (unhandled pixel format): " +
            //    pixeltype.toString() );
            System.Console.WriteLine( "Default (unhandled pixel format): " +
                pixeltype.GetScalarTypeAsString() );
            // Get bytes
            byte[] str1 = new byte[ image.GetBufferLength()];
            image.GetBuffer( str1 );
        }

        return 0;
    }
}

```

27.63 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);
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;
}

```

27.64 GetPortionCSAHeader.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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:

python GetPortionCSAHeader.py input.dcm

Footnote:
    SIEMENS is not publishing any information on the CSA header. So any info
    extracted
    is at your own risk.
"""

import sys
import gdcm

if __name__ == "__main__":

    file = sys.argv[1]

    r = gdcm.Reader()
    r.SetFileName( file )
    if not r.Read():
        sys.exit(1)

    ds = r.GetFile().GetDataSet()
    csa_t1 = gdcm.CSAHeader()
    csa_t2 = gdcm.CSAHeader()
    #print csa
    t1 = csa_t1.GetCSAImageHeaderInfoTag();
    print t1
    t2 = csa_t2.GetCSASeriesHeaderInfoTag();
    print t2
    # Let's do it for t1:
    if ds.FindDataElement( t1 ):
        csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
        print csa_t1

    # Now let's pretend we are only interested in B_value and
    # DiffusionGradientDirection entries:
    bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case
    sensitive !
    print bvalues

    diffgraddir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) #
    WARNING: it is case sensitive !
    print diffgraddir

    # repeat for t2 if you like it:
    if ds.FindDataElement( t2 ):
        csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
        # print csa_t2

    gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
    print gdt

    bv = gdt.GetByteValue();
    #print bv
    str = bv.GetPointer()
    print str.split("\\")

```

27.65 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 = nesteddds.GetDataElement
        ( tX0 );
    const gdcm::DataElement& deY0 = nesteddds.GetDataElement
        ( tY0 );
    const gdcm::DataElement& deX1 = nesteddds.GetDataElement
        ( tX1 );
    const gdcm::DataElement& deY1 = nesteddds.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;
}

```

27.66 GetSubSequenceData.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

/*
 * This example will extract the Movie from the private group of

```

```

* GEMS_Ultrasound_MovieGroup_001 See Attribute
* (7fel,60,GEMS_Ultrasound_MovieGroup_001)
*
* The output file will be stored in 'outvid.dcm' as
* MultiframeGrayscaleByteSecondaryCaptureImageStorage
*/
int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    reader.Read();

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    const PrivateTag tseq(0x7fel,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;
    const DataElement& seq = ds.GetDataElement( tseq );

    SmartPointer<SequenceOfItems> sqi = seq.
        GetValueAsSQ();
    assert( sqi->GetNumberOfItems() == 1 );
    Item &item = sqi->GetItem(1);
    DataSet &subds = item.GetNestedDataSet();

    const PrivateTag tseq1(0x7fel,0x10,"GEMS_Ultrasound_MovieGroup_001"
        );

    if( !subds.FindDataElement( tseq1 ) ) return 1;
    const DataElement& seq1 = subds.GetDataElement( tseq1 );

    SmartPointer<SequenceOfItems> sqi2 = seq1.
        GetValueAsSQ();
    //int n = sqi2->GetNumberOfItems();
    int index = 1;
    Item &item2 = sqi2->GetItem(index);
    DataSet &subds2 = item2.GetNestedDataSet();

    const PrivateTag tseq2(0x7fel,0x20,"GEMS_Ultrasound_MovieGroup_001"
        );

    if( !subds2.FindDataElement( tseq2 ) ) return 1;
    const DataElement& seq2 = subds2.GetDataElement( tseq2 );

    //      std::cout << seq2 << std::endl;

    SmartPointer<SequenceOfItems> sqi3 = seq2.
        GetValueAsSQ();
    int ni3 = sqi3->GetNumberOfItems();
    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();
    int ni6= sqi6->GetNumberOfItems();
    assert( sqi6->GetNumberOfItems() >= 1 );
    const PrivateTag tseq7(0x7fel,0x86,"GEMS_Ultrasound_MovieGroup_001"
        );
    int dimx, dimy;
    for( int i6 = 1; i6 <= ni6; ++i6 )
    {
        Item &item6 = sqi6->GetItem(i6);
        DataSet &subds6 = item6.GetNestedDataSet();

        if( subds6.FindDataElement( tseq7 ) )
        {
            Element<VR::SL, VM::VM4> el;
            el.SetFromDataElement( subds6.GetDataElement
                ( tseq7 ) );
            std::cout << "El= " << el.GetValue() << std::endl;
            dimx = el.GetValue(0);
            dimy = el.GetValue(1);
        }
    }
}

```

```

    }

    const PrivateTag tseq3(0x7fe1,0x36,"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();
    int ni4= sqi4->GetNumberOfItems();
    assert( sqi4->GetNumberOfItems() >= 1 );
    const PrivateTag tseq8(0x7fe1,0x37,"GEMS_Ultrasound_MovieGroup_001"
    );
    const PrivateTag tseq4(0x7fe1,0x43,"GEMS_Ultrasound_MovieGroup_001"
    );
    const PrivateTag tseq5(0x7fe1,0x60,"GEMS_Ultrasound_MovieGroup_001"
    );

    std::vector<char> imbuffer;
    int dimz = 0;
    for( int 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();
#ifdef 0
        {
            std::ofstream out( "/tmp/mo4" );
            out.write( bv4->GetPointer(), bv4->GetLength());
            out.close();
        }
#endif
        const ByteValue *bv5 = seq5.GetByteValue();
#ifdef 0
        {
            std::ofstream out( "/tmp/mo5" );
            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], imbuffer.size() );

    gdcm::SmartPointer<gdcm::Image> im = new
        gdcm::Image;
    im->SetNumberOfDimensions( 3 );

    im->SetDimension(0, dimx );
    im->SetDimension(1, dimy );
    im->SetDimension(2, dimz );
    size_t l1 = imbuffer.size();
    size_t l2 = im->GetBufferLength();
    assert( im->GetBufferLength() == imbuffer.size() );
    im->SetPhotometricInterpretation(
        gdcm::PhotometricInterpretation::MONOCHROME2 );

```

```

im->SetDataElement( fakedata );

gdcm::ImageWriter w;
w.SetImage( *im );
DataSet &dataset = w.GetFile().GetDataSet();

gdcm::UIDGenerator uid;
gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance
    UID
de.SetVR( VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
//ds.Insert( de );
dataset.Replace( de );

de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcm::MediaStorage ms(
    gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage
);
de.SetByteValue( ms.GetString(), strlen(ms.GetString
()));
dataset.Replace( de ); // replace !

w.SetFileName( "outvid.dcm" );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

27.67 headsq2dcm.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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:
python headsq2dcm.py -D /path/to/VTKData
"""

import vtk
import vtkgdcm
from vtk.util.misc import vtkGetDataRoot
VTK_DATA_ROOT = vtkGetDataRoot()

reader = vtk.vtkVolume16Reader()
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)

cast = vtk.vtkImageCast()
cast.SetInput( reader.GetOutput() )
cast.SetOutputScalarTypeToUnsignedChar()

# By default this is creating a Multiframe Grayscale Word Secondary Capture
Image Storage
writer = vtkgdcm.vtkGDCMImageWriter()
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()
```

27.68 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 vtkgdcml;
using Kitware.VTK;
using System;
using System.Runtime.InteropServices;

/*
 * This example shows how vtkgdcml 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)
 *
 * $ MONO_PATH=/home/mmalaterre/Software/
 *   Activiz.NET-5.4.0.455-Linux-x86_64-Personal/bin/ LD_LIBRARY_PATH=/home/mmalaterre/Software/
 *   Activiz.NET-5.4.0.455-Linux-x86_64-Personal/bin:/home/mmalaterre/Projects/gdcm/debug-gcc43/bin/ 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(vtkgdcml.vtkImageData
        imgin)
    {
        IntPtr rawCppThis = imgin.GetCppThis();
        Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis,
            IntPtr.Zero, IntPtr.Zero);
        return imgout;
    }

    static vtkgdcml.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData

```

```

        imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        vtkgdcmm.vtkImageData imgout = new vtkgdcmm.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

    vtkgdcmm.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;
}
}

```

27.69 HelloActiviz2.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.

=====*/
using Kitware.VTK;
using vtkgdcmm;

```

```

/*
 * Usage:
 * export LD_LIBRARY_PATH=/home/mathieu/Perso/gdcm/debug-activiz/bin:/home/
 * mathieu/Software/Activiz.NET-5.4.2.488-Linux-x86_64-Personal/bin
 * export MONO_PATH=/home/mathieu/Software/
 * Activiz.NET-5.4.2.488-Linux-x86_64-Personal/bin
 * $ 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];

        vtkgdc.vtkGDCMImageReader reader = new vtkgdc.vtkGDCMImageReader();
        reader.SetFileName( filename );

        // When calling multiple times creation of C# object from the same C++
        // object it triggers a:
        //error: potential refcounting error: Duplicate rawCppThis - weak reference
        //      that is still alive. Attempting to add '0x00b2dc10' again.
        //      Allowing new wrapped object to take over table key...
        //      Original object should *not* have been destroyed while we still had it
        //      in our table without notifying us...
        //reader.GetOutput();
        //reader.GetOutput();

        System.Console.WriteLine( reader.ToString() ); // Test the ToString compat
        // with Activiz

        vtkGDCMImageWriter writer = new vtkGDCMImageWriter
        ();
        writer.SetInput( reader.GetOutput() );
        writer.SetFileName( outfilename2 );
        writer.Write();

        System.Console.WriteLine( reader.GetOutput().ToString() ); // Test the
        // ToString compat with Activiz

        System.Console.WriteLine( writer.ToString() ); // Test the ToString compat
        // with Activiz

        vtkPNGWriter pngwriter = new vtkPNGWriter();
        pngwriter.SetInput( reader.GetOutput() );
        pngwriter.SetFileName( outfilename );
        pngwriter.Write();

        // at that point the .Write() should have triggered an Update() on the
        // reader:
        if( reader.GetImageFormat() == vtkgdc.vtkgdc.VTK_LUMINANCE ) //
        MONOCHROME2
        {
            System.Console.WriteLine( "Image is MONOCHROME2" ); //
        }

        vtkPNGReader bmpreader = new vtkPNGReader();
        bmpreader.SetFileName( outfilename );

        vtkMedicalImageProperties prop = new vtkMedicalImageProperties();
        prop.SetModality( "MR" );

        vtkMatrix4x4 dircos = reader.GetDirectionCosines();
        dircos.Invert();

        vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter
        ();
        writer2.SetFileName( outfilename2 );
        writer2.SetDirectionCosines( dircos );
        writer2.SetMedicalImageProperties( prop );
        writer2.SetInput( bmpreader.GetOutput() );
        writer2.Write();
    }
}

```

```

    return 0;
}

```

27.70 HelloActiviz3.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
using Kitware.VTK;
using vtkgdcml;

/*
 * $ MONO_PATH=/home/mmalaterre/Software/
   ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/ LD_LIBRARY_PATH=/home/mmalaterre/Software/
   ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/:/home/mmalaterre/Projects/gdcm/debug-gcc43/bin/ 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;
    }
}

```

27.71 HelloActiviz4.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

```

        PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using vtkgdcmm;

/*
 * $ MONO_PATH=/home/mmalaterre/Software/
   ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/ LD_LIBRARY_PATH=/home/mmalaterre/Software/
   ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/:/home/mmalaterre/Projects/gdcm/debug-gcc43/bin/ mono ./bin
   /HelloActiviz4.exe ~/Creatis/gdcmData/test.acr
 *
 */
public class HelloActiviz4
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        vtkGDCMImageReader reader = new vtkGDCMImageReader
        ();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);

        reader.SetFileNames(array);
        reader.Update();

        //System.Console.WriteLine(reader.GetOutput());

        vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

        vtkImageViewer viewer = vtkImageViewer.New();
        viewer.SetInput(reader.GetOutput());
        viewer.SetupInteractor(iren);
        viewer.SetSize(600, 600);
        viewer.Render();

        iren.Initialize();
        iren.Start();

        return 0;
    }
}

```

27.72 HelloActiviz5.cs

```

/*=====
Program: GDICM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed 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 vtkgdcmm;

// 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=/home/mmalaterre/Software/
   ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/
 * $ export LD_LIBRARY_PATH=/home/mmalaterre/Software/
   ActiViz.NET-5.4.0.455-Linux-x86_64-Personal/bin/:/home/mmalaterre/Projects/gdcm/debug-gcc43/bin/
 * $ 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;
}

```

27.73 HelloSimple.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/HelloSimple.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java HelloSimple gdcmData/
   012345.002.050.dcm
 */
import gdcm.*;

public class HelloSimple
{
    public static void main(String[] args) throws Exception
    {
        String filename = args[0];
        Reader reader = new Reader();
        reader.SetFileName( filename );
        boolean ret = reader.Read();
        if( !ret )
        {
            throw new Exception("Could not read: " + filename );
        }
        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();

        System.out.println( ds.toString() );

        System.out.println("Success reading: " + filename );
    }
}

```

27.74 HelloVizWorld.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Basic example for dealing with a DICOM file that contains an Image
 * (read: Pixel Data element)
 */

#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 3 )

```

```

    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    // Instantiate the image reader:
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }
    // If we reach here, we know for sure 2 things:
    // 1. It is a valid DICOM
    // 2. And it contains an Image !

    // The output of superclass gdcm::Reader is a gdcm::File
    //gdcm::File &file = reader.GetFile();

    // The other output of gdcm::ImageReader is a gdcm::Image
    const gdcm::Image &image = reader.GetImage();

    // Let's get some property from the image:
    unsigned int ndim = image.GetNumberOfDimensions();
    // Dimensions of the image:
    const unsigned int *dims = image.GetDimensions();
    // Origin
    const double *origin = image.GetOrigin();
    const gdcm::PhotometricInterpretation &pi =
        image.GetPhotometricInterpretation();
    for(unsigned int i = 0; i < ndim; ++i)
    {
        std::cout << "Dim(" << i << "): " << dims[i] << std::endl;
    }
    for(unsigned int i = 0; i < ndim; ++i)
    {
        std::cout << "Origin(" << i << "): " << origin[i] << std::endl;
    }
    std::cout << "PhotometricInterpretation: " << pi << std::endl;

    // Write the modified DataSet back to disk
    gdcm::ImageWriter writer;
    writer.SetImage( image );
    writer.SetFileName( outfile );
    //writer.SetFile( file ); // We purposely NOT copy the meta information from
        the input
        // file, and instead only pass the image
    if( !writer.Write() )
    {
        std::cerr << "Could not write: " << outfile << std::endl;
        return 1;
    }

    return 0;
}

```

27.75 HelloVTKWorld.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using vtkgdcm;

/*

```

```

* This test only test the SWIG/VTK part, you do not need Activiz
*/
public class HelloVTKWorld
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        vtkGDCMImageReader reader = vtkGDCMImageReader
            .New();
        reader.SetFileName( filename );
        reader.Update();

        vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
        System.Console.WriteLine( prop.GetPatientName() ); //

        if( reader.GetImageFormat() == vtkgdc.vtkgdc.VTK_LUMINANCE ) //
            MONOCHROME2
        {
            System.Console.WriteLine( "Image is MONOCHROME2" ); //
        }

        // Just for fun, invert the direction cosines, output should reflect that:
        vtkMatrix4x4 dircos = reader.GetDirectionCosines();
        dircos.Invert();

        string outfilename = args[1];
        vtkGDCMImageWriter writer = vtkGDCMImageWriter
            .New();
        writer.SetMedicalImageProperties( reader.
            GetMedicalImageProperties() );
        writer.SetDirectionCosines( dircos );
        writer.SetShift( reader.GetShift() );
        writer.SetScale( reader.GetScale() );
        writer.SetImageFormat( reader.GetImageFormat() );
        writer.SetFileName( outfilename );
        //writer.SetInputConnection( reader.GetOutputPort() ); // new
        writer.SetInput( reader.GetOutput() ); // old
        writer.Write();

        return 0;
    }
}

```

27.76 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("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)
{
    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 );
}
}

```

27.77 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;
    }
}

```

27.78 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];

    // Instanciate the reader:
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // If we reach here, we know for sure only 1 thing:
    // It is a valid DICOM file (potentially an old ACR-NEMA 1.0/2.0 file)
    // (Maybe, it's NOT a Dicom image -could be a DICOMDIR, a RTSTRUCT, etc-)

    // The output of gdcm::Reader is a gdcm::File
    gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    gdcm::DataSet &ds = file.GetDataSet();

    // Construct a static(*) type for Image Comments :
    gdcm::Attribute<0x0020,0x4000> imagecomments;
    imagecomments.SetValue( "Hello, World !" );

    // Now replace the Image Comments from the dataset with our:
    ds.Replace( imagecomments.GetAsDataElement() );

    // Write the modified DataSet back to disk
    gdcm::Writer writer;
    writer.CheckFileMetaInformationOff(); // Do not
        attempt to reconstruct the file meta to preserve the file
        // as close to the original as
        possible.
    writer.SetFileName( outfile );
    writer.SetFile( file );
    if( !writer.Write() )
    {
        std::cerr << "Could not write: " << outfile << std::endl;
        return 1;
    }

    return 0;
}

/*
 * (*) static type, means that extra DICOM information VR & VM are computed at
 * compilation time.
 * The compiler is deducing those values from the template arguments of the
 * class.
 */

```

27.79 HelloWorld.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed WITHOUT ANY WARRANTY; without even
#   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#   PURPOSE. See the above copyright notice for more information.
#
#####

"""
Hello World !
"""

```

```

import gdc
import sys

if __name__ == "__main__":

    # verbosity:
    #gdc.Trace.DebugOn()
    #gdc.Trace.WarningOn()
    #gdc.Trace.ErrorOn()

    # Get the filename from the command line
    filename = sys.argv[1]

    # Instantiate a gdc.Reader
    # This is the main class to handle any type of DICOM object
    # You should check for gdc.ImageReader for reading specifically DICOM Image
    file
    r = gdc.Reader()
    r.SetFileName( filename )
    # If the reader fails to read the file, we should stop !
    if not r.Read():
        print "Not a valid DICOM file"
        sys.exit(1)

    # Get the DICOM File structure
    file = r.GetFile()

    # Get the DataSet part of the file
    dataset = file.GetDataSet()

    # Ok let's print it !
    print dataset

    # Use StringFilter to print a particular Tag:
    sf = gdc.StringFilter()
    sf.SetFile(r.GetFile())

    # Check if Attribute exist
    print dataset.FindElement( gdc.Tag(0x0028,0x0010))

    # Let's print it as string pair:
    print sf.ToStringPair(gdc.Tag(0x0028,0x0010))

```

27.80 iU22tomultisc.cxx

```

/*=====
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.

=====*/
/*
 * iU22 Raw Data extractor
 */
#include "gdcReader.h"
#include "gdcImageWriter.h"
#include "gdcAttribute.h"
#include "gdcPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    // IM_001
    const char *filename = argv[1];

    gdc::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, dims[0] );
image.SetDimension(1, dims[1] );
image.SetDimension(2, 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(), 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;
}

```

27.81 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, unsigned int npts, std::vector<double> &
    out )
{
    out.clear();
    for(unsigned int i = 0; i < 2*npts; ++i )
    {
        const unsigned int j = i / 2;
        if( i % 2 )
        {
            if( j != npts - 1 )
            {
                assert( 3*j+5 < 3*npts );
                const double midpointx = (pts[3*j+0] + pts[3*j+3]) / 2;
                const double midpointy = (pts[3*j+1] + pts[3*j+4]) / 2;
                const double midpointz = (pts[3*j+2] + pts[3*j+5]) / 2;
                out.push_back( midpointx );
                out.push_back( midpointy );
                out.push_back( midpointz );
            }
        }
        else
        {
            assert( j < npts );
            out.push_back( pts[3*j+0] );
            out.push_back( pts[3*j+1] );
            out.push_back( pts[3*j+2] );
        }
    }
    assert( out.size() == 2 * npts * 3 - 3 );
    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    gdcm::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( changeprivatetags );

```

```

fef.SetFile( reader.GetFile() );
if( !fef.Change() )
{
    std::cerr << "Failed to change: " << filename << std::endl;
    return 1;
}

// (3006,0039) SQ (Sequence with undefined length #=4)      # u/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( out.size() / 3 );
at_interpolate.SetValues( &out[0], 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, 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;
}

```

27.82 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();
cast->SetInput( reader->GetOutput() );
cast->SetOutputScalarTypeToUnsignedShort();

vtkImageMagnify *magnify = vtkImageMagnify::New();
magnify->SetInput( cast->GetOutput() );
magnify->SetInterpolate( 1 );
magnify->SetInterpolate( 0 );
int factor = 100;
magnify->SetMagnificationFactors (factor, factor, 1);

vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New
();
writer->SetFileName( "/tmp/bla.dcm" );
writer->SetInput( magnify->GetOutput() );
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;
}

```

27.83 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;
}
}

```

27.84 ManipulateFile.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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:

python ManipulateFile.py input.dcm output.dcm

Footnote:
GDCM 1.2.x would create incorrect Multiframe MR Image Storage file. Try to
recover from
the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage
file.
e.g:

python ManipulateFile.py
    Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
"""

import sys
import gdcm

if __name__ == "__main__":

    file1 = sys.argv[1]
    file2 = sys.argv[2]

    r = gdcm.Reader()
    r.SetFileName( file1 )
    if not r.Read():
        sys.exit(1)

    ano = gdcm.Anonymizer()
    ano.SetFile( r.GetFile() )
    ano.RemovePrivateTags()
    ano.Remove( gdcm.Tag(0x0032,0x1030) )
    ano.Remove( gdcm.Tag(0x008,0x14) )
    ano.Remove( gdcm.Tag(0x008,0x1111) )
    ano.Remove( gdcm.Tag(0x008,0x1120) )
    ano.Remove( gdcm.Tag(0x008,0x1140) )
    ano.Remove( gdcm.Tag(0x10,0x21b0) )
    ano.Empty( gdcm.Tag(0x10,0x10) )

```

```

ano.Empty( gdc.Tag(0x10,0x20) )
ano.Empty( gdc.Tag(0x10,0x30) )
ano.Empty( gdc.Tag(0x20,0x10) )
ano.Empty( gdc.Tag(0x32,0x1032) )
ano.Empty( gdc.Tag(0x32,0x1033) )
ano.Empty( gdc.Tag(0x40,0x241) )
ano.Empty( gdc.Tag(0x40,0x254) )
ano.Empty( gdc.Tag(0x40,0x253) )
ano.Empty( gdc.Tag(0x40,0x1001) )
ano.Empty( gdc.Tag(0x8,0x80) )
ano.Empty( gdc.Tag(0x8,0x50) )
ano.Empty( gdc.Tag(0x8,0x1030) )
ano.Empty( gdc.Tag(0x8,0x103e) )
ano.Empty( gdc.Tag(0x18,0x1030) )
ano.Empty( gdc.Tag(0x38,0x300) )
g = gdc.UIDGenerator()
ano.Replace( gdc.Tag(0x0008,0x0018), g.Generate() )
ano.Replace( gdc.Tag(0x0020,0x00d), g.Generate() )
ano.Replace( gdc.Tag(0x0020,0x00e), g.Generate() )
ano.Replace( gdc.Tag(0x0020,0x052), g.Generate() )
#ano.Replace( gdc.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
"""
ano.Remove( gdc.Tag(0x0018,0x0020) ) # ScanningSequence
ano.Remove( gdc.Tag(0x0018,0x0021) ) # SequenceVariant
ano.Remove( gdc.Tag(0x0018,0x0022) ) # ScanOptions
ano.Remove( gdc.Tag(0x0018,0x0023) ) # MRACquisitionType
ano.Remove( gdc.Tag(0x0018,0x0050) ) # SliceThickness
ano.Remove( gdc.Tag(0x0018,0x0080) ) # RepetitionTime
ano.Remove( gdc.Tag(0x0018,0x0081) ) # EchoTime
ano.Remove( gdc.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
ano.Remove( gdc.Tag(0x0018,0x0091) ) # EchoTrainLength
ano.Remove( gdc.Tag(0x0018,0x1164) ) # ImagerPixelSpacing

ano.Remove( gdc.Tag(0x0020,0x0032) ) # Image Position (Patient)
ano.Remove( gdc.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
ano.Remove( gdc.Tag(0x0020,0x0052) ) # Frame of Reference UID
ano.Remove( gdc.Tag(0x0020,0x1040) ) # Position Reference Indicator

ano.Replace( gdc.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation

ano.Empty( gdc.Tag(0x0020,0x0020) )

ano.Remove( gdc.Tag(0x7fe0,0x0000) )

#ano.Empty( gdc.Tag(0x0028,0x0009) ) # Frame Increment Pointer

#ano.Empty( gdc.Tag(0x0028,0x1052) ) #<entry group="0028" element="1052"
#vr="DS" vm="1" name="Rescale Intercept"/>
#ano.Empty( gdc.Tag(0x0028,0x1053) ) #<entry group="0028" element="1053"
#vr="DS" vm="1" name="Rescale Slope"/>
#ano.Replace( gdc.Tag(0x0028,0x1054), "US" ) #<entry group="0028" element="
1054" vr="LO" vm="1" name="Rescale Type"/>

ano.Replace( gdc.Tag(0x2050, 0x0020), "IDENTITY")
"""

w = gdc.Writer()
w.SetFile( ano.GetFile() )
w.SetFileName( file2 )
if not w.Write():
    sys.exit(1)

```

27.85 ManipulateSequence.py

```

#####
#
# 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.
#
#####

```

```

"""
Usage:

python ManipulateSequence.py input.dcm output.dcm

This was tested using:

python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

This is a dummy example on how to modify a value set in a nested-nested dataset

WARNING:
Do not use as-is in production, this is just an example
This example works in an undefined length Item only (you need to explicitly
    recompute the length otherwise)
"""

import sys
import gdcm

if __name__ == "__main__":

    file1 = sys.argv[1]
    file2 = sys.argv[2]

    r = gdcm.Reader()
    r.SetFileName( file1 )
    if not r.Read():
        sys.exit(1)

    f = r.GetFile()
    ds = f.GetDataSet()
    tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
    if ds.FindDataElement( tsis ):
        sis = ds.GetDataElement( tsis )
        #sqsis = sis.GetSequenceOfItems()
        # GetValueAsSQ handle more cases
        sqsis = sis.GetValueAsSQ()
        if sqsis.GetNumberOfItems():
            item1 = sqsis.GetItem(1)
            nestedds = item1.GetNestedDataSet()
            tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
            if nestedds.FindDataElement( tprcs ):
                prcs = nestedds.GetDataElement( tprcs )
                sqprcs = prcs.GetSequenceOfItems()
                if sqprcs.GetNumberOfItems():
                    item2 = sqprcs.GetItem(1)
                    nestedds2 = item2.GetNestedDataSet()
                    # (0008,0104) LO [Uncompressed predecessor] # 24, 1
                    CodeMeaning
                    tcm = gdcm.Tag(0x0008,0x0104)
                    if nestedds2.FindDataElement( tcm ):
                        cm = nestedds2.GetDataElement( tcm )
                        mystr = "GDCM was here"
                        cm.SetByteValue( mystr, gdcm.VL( len(mystr) ) )

    w = gdcm.Writer()
    w.SetFile( f )
    w.SetFileName( file2 )
    if not w.Write():
        sys.exit(1)

```

27.86 MergeFile.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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:

python MergeFile.py input1.dcm input2.dcm

It will produce a 'merge.dcm' output file, which contains all meta
information from input1.dcm
and copy the Stored Pixel values from input2.dcm
This script even works when input2.dcm is a Secondary Capture and does not
contains information
such as IOP and IPP...
"""

import sys
import gdc

if __name__ == "__main__":

    file1 = sys.argv[1]
    file2 = sys.argv[2]

    r1 = gdc.ImageReader()
    r1.SetFileName( file1 )
    if not r1.Read():
        sys.exit(1)

    r2 = gdc.ImageReader()
    r2.SetFileName( file2 )
    if not r2.Read():
        sys.exit(1)

    # Image from r2 could be Secondary Capture and thus would not contains
    # neither IPP nor IOP
    # Instead always prefer to only copy the Raw Data Element.
    # Warning ! Image need to be identical ! Only the value of Stored Pixel can
    # be different.
    r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )

    w = gdc.ImageWriter()
    w.SetFile( r1.GetFile() )
    #w.SetImage( r2.GetImage() ) # See comment above
    w.SetImage( r1.GetImage() )

    w.SetFileName( "merge.dcm" )
    if not w.Write():
        sys.exit(1)

    sys.exit(0)
```

27.87 MergeTwoFiles.cxx

```
/*=====
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 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 gdcData/012345.002.050.dcm gdcData/test.acr merge.dcm
 */

#include "gdcReader.h"
#include "gdcImageReader.h"
#include "gdcImageWriter.h"
```

```

#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *file1 = argv[1];
    const char *file2 = argv[2];
    const char *file3 = argv[3];

    // Read file1
    gdcm::ImageReader reader1;
    reader1.SetFileName( file1 );
    if( !reader1.Read() )
    {
        return 1;
    }

    // Read file2
    gdcm::ImageReader reader2;
    reader2.SetFileName( file2 );
    if( !reader2.Read() )
    {
        return 1;
    }

    // Ok now let's take the DataSet from file1 and the Image from file2
    // Warning: if file2 is -for example- a Secondary Capture Storage, then it
    // has no
    // Image Orientation (Patient) thus any Image Orientation (Patient) from
    // file1
    // will be discarded...

    // let's be fancy. In case reader2 contains explicit, but reader1 is implicit
    // we would rather see an implicit output
    if( reader1.GetFile().GetHeader().GetDataSetTransferSyntax
        () == gdcm::TransferSyntax::ImplicitVRLittleEndian
        )
    {
        reader2.GetImage().SetTransferSyntax(
            gdcm::TransferSyntax::ImplicitVRLittleEndian
        );
    }

    gdcm::ImageWriter writer;
    writer.SetFileName( file3 );
    writer.SetFile( reader1.GetFile() );
    // ImageWriter will always use all of gdcm::Image information an override
    // anything wrong from
    // reader1.GetFile(), including the Transfer Syntax
    writer.SetImage( reader2.GetImage() );

    gdcm::DataSet &ds = reader1.GetFile().GetDataSet
        ();

    // Make sure that SOPInstanceUID are different
    // Simply removing it is sufficient as gdcm::ImageWriter will generate one by
    // default
    // if not found.
    ds.Remove( gdcm::Tag(0x0008,0x0018) );
    if( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```

27.88 MetalmageMD5Activiz.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 vtkgdcm;
using gdcm;

/*
 * $ export MONO_PATH=/usr/lib/cli/activiz-cil/
 * $ export LD_LIBRARY_PATH=/usr/lib/cli/activiz-cil/
 * $ 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.WriteLine( "Problem with raw file: " + filename + "\n" );
        System.Console.WriteLine( digestraw );
        System.Console.WriteLine( "\n" );
        System.Console.WriteLine( rawref );
        System.Console.WriteLine( "\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;
}
}

```

27.89 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();
}
}

```

27.90 MPRViewer.java

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.

```

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdc.jar:/usr/share/java/vtk.jar javac MPRViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/
 * jni:. CLASSPATH=/usr/share/java/vtk.jar:vtkgdc.jar:gdcm.jar:. java MPRViewer BRAINX
 */
public class MPRViewer
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        // VTK-GDCM
        System.loadLibrary("vtkgdc.jar");
    }

    static FilenamesType fns = new FilenamesType();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
        else
        {
            process(dir.getPath());
        }
    }

    public static void main(String[] args) throws Exception
    {
        String dirname = args[0];
        if( !PosixEmulation.FileIsDirectory( dirname ) )
        {
            return;
        }

        File dir = new File(dirname);
        visitAllFiles(dir);

        IPPSorter ipp = new IPPSorter();
        ipp.SetComputeZSpacing( true );
        ipp.SetZSpacingTolerance( 1e-3 );
        boolean b = ipp.Sort( fns );
        if(!b)
        {
            throw new Exception("Could not scan");
        }
        double ippzspacing = ipp.GetZSpacing();

        FilenamesType sorted = ipp.GetFilenames();
        vtkStringArray files = new vtkStringArray();

```

```

long nfiles = sorted.size();
//for( String f : sorted )
for (int i = 0; i < nfiles; i++) {
    String f = sorted.get(i);
    files.InsertNextValue( f );
}
vtkGDCMImageReader reader = new vtkGDCMImageReader
();
reader.SetFileNames( files );
reader.Update(); // get spacing value

double[] spacing = reader.GetOutput().GetSpacing();

vtkImageChangeInformation change = new vtkImageChangeInformation();
change.SetInputConnection( reader.GetOutputPort() );
change.SetOutputSpacing( spacing[0], spacing[1], ipzspacing );

// 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();
}
}

```

27.91 MPRViewer2.java

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.

```

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer2.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/
  jni:. CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MPRViewer2
  BRAINX
 *
 */
public class MPRViewer2
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        System.loadLibrary("vtkHybridJava");
        System.loadLibrary("vtkWidgetsJava");
        // VTK-GDCM
        System.loadLibrary("vtkgdcmJava");
    }

    static FilenamesType fns = new FilenamesType();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
        else
        {
            process(dir.getPath());
        }
    }

    public void dointer(vtkImagePlaneWidget current_widget)
    {
        int cstat = current_widget.GetCursorDataStatus();
        double[] v = current_widget.GetCurrentCursorPosition();
        //System.out.println( cstat );
        //System.out.println( v[0] );
        //System.out.println( v[1] );
        //System.out.println( v[2] );
        planeWidgetX.SetSliceIndex( (int)v[0] );
        planeWidgetY.SetSliceIndex( (int)v[1] );
        planeWidgetZ.SetSliceIndex( (int)v[2] );
        planeWidgetX.GetCurrentRenderer().ResetCameraClippingRange();
        planeWidgetY.GetCurrentRenderer().ResetCameraClippingRange();
        planeWidgetZ.GetCurrentRenderer().ResetCameraClippingRange();
    }

    public void startinterX()
    {
        dointer( planeWidgetX );
    }

    public void interX()

```

```

        {
            dointer( planeWidgetX );
        }
    public void endinterX()
    {
    }
    public void startinterY()
    {
        dointer( planeWidgetY );
    }
    public void interY()
    {
        dointer( planeWidgetY );
    }
    public void endinterY()
    {
    }
    public void startinterZ()
    {
        dointer( planeWidgetZ );
    }
    public void interZ()
    {
        dointer( planeWidgetZ );
    }
    public void endinterZ()
    {
        //System.out.println( "endinter" );
    }

    public static void AlignCamera(int slice_number, vtkImagePlaneWidget
        current_widget)
    {
        vtkImageData image = (vtkImageData)current_widget.GetInput();
        vtkRenderer ren = current_widget.GetCurrentRenderer();
        double[] origin = image.GetOrigin();
        double ox = origin[0];
        double oy = origin[1];
        double oz = origin[2];

        int 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], ippzspacing );
    change.Update();

    System.out.println( change.GetOutput().toString() );

    vtkRenderer ren1 = new vtkRenderer();
    ren1.SetViewport(0., 0., 0.333, 1);
    ren1.SetBackground(0.1,0.2,0.4);
    vtkRenderer ren2 = new vtkRenderer();
    ren2.SetViewport(0.333, 0., 0.667, 1);
    ren2.SetBackground(0.1,0.2,0.4);
    vtkRenderer ren3 = new vtkRenderer();
    ren3.SetViewport(0.667, 0., 1., 1.);
    ren3.SetBackground(0.1,0.2,0.4);

    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);
    renWin.AddRenderer(ren2);
    renWin.AddRenderer(ren3);

    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
    iren.SetRenderWindow(renWin);

    vtkInteractorStyleImage style = new vtkInteractorStyleImage();
    iren.SetInteractorStyle( style );

    vtkCellPicker picker = new vtkCellPicker();
    picker.SetTolerance(0.005);

    vtkProperty ipwProp = new vtkProperty();

    //vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();

```

```

planeWidgetX.SetInteractor(iren);
planeWidgetX.SetCurrentRenderer(ren1);
planeWidgetX.SetDefaultRenderer(ren1);
planeWidgetX.RestrictPlaneToVolumeOn();
planeWidgetX.SetTexturePlaneProperty(ipwProp);
//planeWidgetX.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetX.TextureInterpolateOff();
//planeWidgetX.SetResliceInterpolateToNearestNeighbour();
planeWidgetX.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 );
}
}

```

27.92 MrProtocol.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.

=====*/
/*
 *
 */

/*
28 - 'MrProtocol' VM 1, VR UN, SyngoDT 0, NoOfItems 6, Data '### ASCCONV BEGIN
    ###
ulVersion                               = 0xb332
tSequenceFileName                       = "%SiemensSeq%\flfq_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.ulEnableRFSpooling = 0x1
sPhysioImaging.lSignal1      = 2
sPhysioImaging.lMethod1      = 2
sPhysioImaging.lSignal2      = 1
sPhysioImaging.lMethod2      = 1
sPhysioImaging.lPhases       = 21
sPhysioImaging.lRetroGatedImages = 16
sPhysioImaging.sPhysioECG.lScanWindow = 805
sPhysioImaging.sPhysioECG.lTriggerPulses = 1
sPhysioImaging.sPhysioECG.lTriggerWindow = 5
sPhysioImaging.sPhysioECG.lArrhythmiaDetection = 1
sPhysioImaging.sPhysioECG.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioECG.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioPulse.lTriggerPulses = 1
sPhysioImaging.sPhysioPulse.lTriggerWindow = 5
sPhysioImaging.sPhysioPulse.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioPulse.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioExt.lTriggerPulses = 1
sPhysioImaging.sPhysioExt.lTriggerWindow = 5
sPhysioImaging.sPhysioExt.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioExt.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioResp.lRespGateThreshold = 20
sPhysioImaging.sPhysioResp.lRespGatePhase = 2
sPhysioImaging.sPhysioResp.dGatingRatio = 0.3
sSpecPara.lPhaseCyclingType = 1
sSpecPara.lPhaseEncodingType = 1
sSpecPara.lRFExcitationBandwidth = 1
sSpecPara.ucRemoveOversampling = 0x1
sSpecPara.lDecouplingType = 1
sSpecPara.lNOEType = 1
sSpecPara.lExcitationType = 1
sSpecPara.lSpectralSuppression = 1
sDiffusion.ulMode = 0x1
sAngio.sFlowArray.asElm[0].nVelocity = 100
sAngio.sFlowArray.asElm[0].nDir = 0x4
sAngio.sFlowArray.lSize = 1
sAngio.ucPCFlowMode = 0x2
sAngio.ucTOFInflow = 0x4
sAngio.ucRephasedImage = 0x1
sAngio.ucPhaseImage = 0x1
sEllipticalFilter.ucMode = 0x1
sPat.lAccelFactPE = 1
sPat.lAccelFact3D = 1
sPat.ucPATMode = 0x1
sPat.ucRefScanMode = 0x1
ucAutoMovie = 0x1
ucDisableChangeStoreImages = 0x1
ucReconstructionMode = 0x1
ucPHAPSMode = 0x1
ucDixon = 0x1
lAverages = 2
adFlipAngleDegree[0] = 30
lScanTimeSec = 103
lTotalScanTimeSec = 112
dRefSNR = 165404.1473
dRefSNR_VOI = 165404.1473
tdefaultEVAProt =
    "%SiemensEvaDefProt%\Inline\Inline.evp"
tcurrentEVAProt = "%CURRENTEVAPROT%\EVA2A5.tmp"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tElement = "PP6"
sCOIL_SELECT_MEAS.asList[0].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[0].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tElement = "PP5"
sCOIL_SELECT_MEAS.asList[1].lElementSelected = 1

```

```

sCOIL_SELECT_MEAS.asList[1].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tElement = "PP3"
sCOIL_SELECT_MEAS.asList[2].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[2].lRxChannelConnected = 2
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tElement = "PP4"
sCOIL_SELECT_MEAS.asList[3].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[3].lRxChannelConnected = 3
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tElement = "PP2"
sCOIL_SELECT_MEAS.asList[4].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[4].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tElement = "PP1"
sCOIL_SELECT_MEAS.asList[5].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[5].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tElement = "PA6"
sCOIL_SELECT_MEAS.asList[6].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[6].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tElement = "PA5"
sCOIL_SELECT_MEAS.asList[7].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[7].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tElement = "PA3"
sCOIL_SELECT_MEAS.asList[8].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[8].lRxChannelConnected = 6
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tElement = "PA4"
sCOIL_SELECT_MEAS.asList[9].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[9].lRxChannelConnected = 7
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tElement = "PA2"
sCOIL_SELECT_MEAS.asList[10].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[10].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tElement = "PA1"
sCOIL_SELECT_MEAS.asList[11].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[11].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[0] = 0xff
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[1] = 0x76
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[2] = 0x78
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[3] = 0x87
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[4] = 0x67
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[0] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[1] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[2] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[3] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[4] = 0x2
sEFISPEC.bEFIDataValid = 1
### ASCCONV END ###
,
*/

/*
 * Table of equivalence:
 *
ulVersion = 0xbee332
<=>
27 - 'MrProtocolVersion' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '12510002'
*/

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"

```

```

#include <map>

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }

    gdcm::CSAHeader csa;
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet
        ();

    //const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
    const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag
        ();

    if( ds.FindDataElement( t2 ) )
    {
        csa.LoadFromDataElement( ds.GetDataElement
            ( t2 ) );
        //csa.Print( std::cout );
    }

    if( !csa.FindCSAElementByName( "MrProtocol" ) )
    {
        return 1;
    }
    const gdcm::CSAElement &csael = csa.GetCSAElementByName
        ( "MrProtocol" );
    //std::cout << csael << std::endl;

    const gdcm::ByteValue *bv = csael.GetByteValue();
    if( !bv )
    {
        return 1;
    }
    std::string str(bv->GetPointer(), bv->GetLength());
    std::istringstream is(str);
    std::string s;
    typedef std::map< std::string, std::string > MyMapType;
    MyMapType mymap;
    while( std::getline(is, s) )
    {
        std::string::size_type pos = s.find( '=' );
        if( pos != std::string::npos )
        {
            std::string sub1 = s.substr(0, pos);
            sub1.erase( sub1.find_last_not_of(' ') + 1);
            std::string sub2 = s.substr(pos+1); // skip the '=' char
            sub2.erase( 0, sub2.find_first_not_of(' '));
            //std::cout << sub1 << std::endl;
            mymap.insert( MyMapType::value_type(sub1, sub2) );
        }
        else
        {
            // ### ASCCONV BEGIN ###
            // ### ASCCONV END ###
        }
    }

    const char fourierstr[] = "sKSpace.ucSlicePartialFourier";
    const gdcm::CSAHeaderDict &csadict =
        gdcm::Global::GetInstance().GetDicts().
        GetCSAHeaderDict();
    const gdcm::CSAHeaderDictEntry &fourier = csadict.
        GetCSAHeaderDictEntry( fourierstr );
    std::cout << fourier << std::endl;
    MyMapType::const_iterator it = mymap.find ( fourierstr );
    if( it == mymap.end() ) return 1;
    //std::cout << it->second << std::endl;
    const std::string &partial_fourier = it->second;
    if( partial_fourier == "0x1" )
    {

```

```

    std::cout << "partial fourier is 4/8" << std::endl;
}
else if( partial_fourier == "0x2" )
{
    std::cout << "partial fourier is 5/8" << std::endl;
}
else if( partial_fourier == "0x4" )
{
    std::cout << "partial fourier is 6/8" << std::endl;
}
else if( partial_fourier == "0x8" )
{
    std::cout << "partial fourier is 7/8" << std::endl;
}
else if( partial_fourier == "0x10" )
{
    std::cout << "partial fourier is 8/8" << std::endl;
}
else
{
    std::cerr << "Impossible: " << partial_fourier << std::endl;
    return 1;
}
}

/*
This is the Flip Angle:
adFlipAngleDegree[0] = 30

One can find it also in the protocol:

...
    <ParamFuncion>."TlmapFuncion">
    {
        <Class> "TlmapFuncion@IceImagePostProcFuncion">

        <ParamBool>."EXECUTE"> { }
        <ParamDouble>."Flipl_deg"> { <Precision> 16 14.7378520000000000 }
    }
...

*/
// Below is an attempt to play with the CSAHeader dict:
#ifdef 0
const char gspec[] = "sGRADSPEC.flSensitivityX";
it = mymap.find( gspec );
if( it == mymap.end() ) return 1;
const std::string &dummy = it->second;
std::cout << dummy << std::endl;

const gdcm::CSAHeaderDictEntry &csaentry = csadict.
    GetCSAHeaderDictEntry( gspec );
std::cout << csaentry << std::endl;
#endif

/*
sSliceArray.ucMode -- should be in (1, 2, 4)
enum SeriesMode
{
    ASCENDING = 0x01,
    DESCENDING = 0x02,
    INTERLEAVED = 0x04
};
*/
const char sliceorderstr[] = "sSliceArray.ucMode";
const gdcm::CSAHeaderDictEntry &sliceorder = csadict.
    GetCSAHeaderDictEntry( sliceorderstr );
std::cout << sliceorder << std::endl;

it = mymap.find( sliceorderstr );
if( it == mymap.end() ) return 1;
const std::string &slice_order = it->second;
if( slice_order == "0x1" )
{
    std::cout << "slice_order: ASCENDING" << std::endl;
}
else if( slice_order == "0x2" )
{
    std::cout << "slice_order: DESCENDING" << std::endl;
}
else if( slice_order == "0x4" )
{
    std::cout << "slice_order: INTERLEAVED" << std::endl;
}

```

```

    }
    else
    {
        std::cerr << "Impossible: " << slice_order << std::endl;
        return 1;
    }

    return 0;
}

```

27.93 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;
}
}

```

27.94 NewSequence.py

```

#####
#
# Program: GDCM (Grassroots DICOM). A DICOM library
#
# Copyright (c) 2006-2011 Mathieu Malaterre
# All rights reserved.
# See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
#
# This software is distributed 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:

    python NewSequence.py input.dcm output.dcm

Thanks to Robert Irie for code
"""

import sys
import gdcM

if __name__ == "__main__":

    file1 = sys.argv[1]
    file2 = sys.argv[2]

    r = gdcM.Reader()
    r.SetFileName( file1 )
    if not r.Read():
        sys.exit(1)

    f = r.GetFile()
    ds = f.GetDataSet()
    #tsis = gdcM.Tag(0x0008,0x2112) # SourceImageSequence

    # Create a dataelement
    de = gdcM.DataElement(gdcM.Tag(0x0010, 0x2180))
    de.SetByteValue("Occupation", gdcM.VL(len("Occupation")))
    de.SetVR(gdcM.VR(gdcM.VR.SH))

    # Create an item
    it=gdcM.Item()
    it.SetVLToUndefined() # Needed to not popup error message
    #it.InsertDataElement(de)
    nds=it.GetNestedDataSet()
    nds.Insert(de)

    # Create a Sequence
    sq=gdcM.SequenceOfItems().New()
    sq.SetLengthToUndefined()
    sq.AddItem(it)

    # Insert sequence into data set

```

```

des=gdcmm.DataElement(gdcmm.Tag(0x0400,0x0550))
des.SetVR(gdcmm.VR(gdcmm.VR.SQ))
des.SetValue(sq.__ref__())
des.SetVLToUndefined()

ds.Insert(des)

w = gdcmm.Writer()
w.SetFile( f )
w.SetFileName( file2 )
if not w.Write():
    sys.exit(1)

```

27.95 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();
    windowLevel->SetInput( reader->GetOutput() );
    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();
    actor->SetInput( windowLevel->GetOutput() );

    renderer->AddActor( actor );

    renWin->Render();
}

```

```

vtkWindowToImageFilter *w2if = vtkWindowToImageFilter::New();
w2if->SetInput ( renWin );

vtkPNGWriter *wr = vtkPNGWriter::New();
wr->SetInput( w2if->GetOutput() );
wr->SetFileName ( "offscreenimage.png" );
wr->Write();

reader->Delete();
renWin->Delete();
renderer->Delete();
windowlevel->Delete();
actor->Delete();
w2if->Delete();
wr->Delete();

return 0;
}

```

27.96 PatchFile.cxx

This is a C++ example on how to use `gdcm::Attribute`

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * The image was a broken file where the Pixel Data element was 8 times too big
 * Apparently multiplying the BitsAllocated to 4 and multiplying the number of
 * frames by 2 would solve the problem
 *
 * This C++ code can be used to patch the header.
 */

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *f = argv[1];
    const char *out = argv[2];
    gdcm::Reader r;
    r.SetFileName( f );
    if( !r.Read() )
    {
        return 1;
    }

    gdcm::File &file = r.GetFile();
    gdcm::DataSet& ds = file.GetDataSet();
    // (0028,0100) US 16                                     # 2, 1
    // BitsAllocated
    // (0028,0101) US 16                                     # 2, 1
    // BitsStored
    // (0028,0102) US 15                                     # 2, 1 HighBit
    //
    {
        gdcm::Attribute<0x28,0x100> at;
        at.SetFromDataElement( ds.GetDataElement(

```

```

        at.GetTag() );
    if( at.GetValue() != 8 )
    {
        return 1;
    }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
    {
        gdcm::Attribute<0x28,0x101> at;
        at.SetFromDataElement( ds.GetDataElement(
            at.GetTag() ) );
        if( at.GetValue() != 8 )
        {
            return 1;
        }
        at.SetValue( 32 );
        ds.Replace( at.GetAsDataElement() );
    }
    {
        gdcm::Attribute<0x28,0x102> at;
        at.SetFromDataElement( ds.GetDataElement(
            at.GetTag() ) );
        if( at.GetValue() != 7 )
        {
            return 1;
        }
        at.SetValue( 31 );
        ds.Replace( at.GetAsDataElement() );
    }
    // (0028,0008) IS [56] # 2, 1
    NumberOfFrames

    {
        gdcm::Attribute<0x28,0x8> at;
        at.SetFromDataElement( ds.GetDataElement(
            at.GetTag() ) );
        at.SetValue( at.GetValue() * 2 );
        ds.Replace( at.GetAsDataElement() );
    }

    gdcm::Writer w;
    w.SetFile( file );
    w.SetCheckFileMetaInformation( false );
    w.SetFileName( out );
    if( !w.Write() )
    {
        return 1;
    }

    // Now let's see if we can read it as an image:
    gdcm::ImageReader ir;
    ir.SetFileName( out );
    if(!ir.Read())
    {
        return 1;
    }
    gdcm::Image &image = ir.GetImage();
    unsigned long len = image.GetBufferLength();
    const gdcm::ByteValue *bv = ir.GetFile().GetDataSet
        ().GetDataElement( gdcm::Tag(0x7fe0,0x0010) ).
        GetByteValue();
    if( !bv || len != bv->GetLength() )
    {
        return 1;
    }
    std::cout << bv->GetLength() << " " << len << std::endl;

    std::cout << "Success to rewrite image !" << std::endl;
    image.Print( std::cout );
    return 0;
}

```

27.97 PhilipsPrivateRescaleInterceptSlope.py

```

#####
#

```

```

# Program: GDCM (Grassroots DICOM). A DICOM library
#
# Copyright (c) 2006-2011 Mathieu Malaterre
# All rights reserved.
# See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
# This software is distributed 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:

python

import gdcm
import sys

filename = sys.argv[1]
tmpfile = "/tmp/philips_rescaled.dcm"

# Need to access some private tags, read the file :
reader = gdcm.Reader()
reader.SetFileName( filename )
if not reader.Read():
    sys.exit(1)

ds = reader.GetFile().GetDataSet()

#print ds
# (2005,1409)      DS      4      0.0
# (2005,140a)      DS     16     1.52283272283272

# (2005,0014)      LO      26      Philips MR Imaging DD 005
tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
print tag1
print tag2

# make sure to do a copy, we want the private tag to remain
# otherwise gdcm gives us a reference
e11 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
print e11
e12 = gdcm.DataElement( ds.GetDataElement( tag2 ) )
print e12

# (0028,1052) DS [-1000]                                # 6, 1
# RescaleIntercept
# (0028,1053) DS [1]                                     # 2, 1 RescaleSlope

e11.SetTag( gdcm.Tag(0x0028,0x1052) )
e12.SetTag( gdcm.Tag(0x0028,0x1053) )

ds.Insert( e11 )
ds.Insert( e12 )

w = gdcm.Writer()
w.SetCheckFileMetaInformation( False )
w.SetFileName( tmpfile )
w.SetFile( reader.GetFile() )
if not w.Write():
    sys.exit(1)

print "success"

```

27.98 PlaySound.py

```

#####
#
# Program: GDCM (Grassroots DICOM). A DICOM library
#
# Copyright (c) 2006-2011 Mathieu Malaterre
# All rights reserved.

```

```

# See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
# This software is distributed 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:

    python PlaySound.py input.dcm
"""

import gdcm
import sys

#filename =
#    "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/audio_from_rafael_sanguinetti.dcm"
filename = sys.argv[1]
print filename

r = gdcm.Reader()
r.SetFileName( filename )
if not r.Read():
    sys.exit(1)

ds = r.GetFile().GetDataSet()

waveformtag = gdcm.Tag(0x5400,0x0100)
waveformsq = ds.GetDataElement( waveformtag )
#print waveformsq

#print dir(waveformsq)

items = waveformsq.GetSequenceOfItems()

if not items.GetNumberOfItems():
    sys.exit(1)

item = items.GetItem(1)
#print item

waveformds = item.GetNestedDataSet()
#print waveformds

waveformdatatag = gdcm.Tag(0x5400,0x1010)
waveformdata = waveformds.GetDataElement( waveformdatatag )

#print waveformdata.GetPointer()
bv = waveformdata.GetByteValue()
print dir(bv)

#print bv.GetPointer()
print bv.GetLength()
l = 116838

file='test.wav'
myfile = open(file, "wb")
s = bv.GetPointer()
for i in range(0, l):
    myfile.write(s[i])
myfile.close()

# http://mail.python.org/pipermail/python-list/2004-October/288905.html
if sys.platform.startswith('win'):
    from winsound import PlaySound, SND_FILENAME, SND_ASYNC
    PlaySound(file, SND_FILENAME|SND_ASYNC)
elif sys.platform.find('linux')>-1:
    from wave import open as waveOpen
    from ossaudiodev import open as ossOpen
    s = waveOpen(file,'rb')
    (nc,sw,fr,nf,comptype, compname) = s.getparams( )
    dsp = ossOpen('/dev/dsp','w')
    try:
        from ossaudiodev import AFMT_S16_NE
    except ImportError:
        if byteorder == "little":
            AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
        else:
            AFMT_S16_NE = ossaudiodev.AFMT_S16_BE

```

```
dsp.setparameters(AFMT_S16_NE, nc, fr)
data = s.readframes(nf)
s.close()
dsp.write(data)
dsp.close()
```

27.99 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() );

gdcmm::DataElement pixeldata( gdcmm::Tag(0x7fe0,0
    x0010) );
pixeldata.SetVR( gdcmm::VR::OW );
pixeldata.SetByteValue( (char*)&buffer[0], 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
(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian
);
gdcmm::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 gdcmm::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;
}

```

27.100 PrivateDict.py

```

#####
#
#   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.
#
#####

"""
"""

import gdcmm
import sys,os

if __name__ == "__main__":
    #gdcmm.Trace.DebugOn()
    globInst = gdcmm.Global.GetInstance()
    # Try to load Part3.xml file
    # This file is too big for being accessible directly at runtime.
    globInst.LoadResourcesFiles()

```

```
# Get a private tag from the runtime dicts. LoadResourcesFiles could
# have failed but this has no impact on the private dict

d = globInst.GetDicts()
print d.GetDictEntry( gdcM.Tag(0x0029,0x0010) , "SIEMENS CSA HEADER" )
pd = d.GetPrivateDict()
print pd.GetDictEntry( gdcM.PrivateTag(0x0029,0x0010,"SIEMENS
    CSA HEADER" ) )
```

27.101 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:
gdcmm::PrivateTag dummy = private_tag;
dummy.SetElement( 0x1000 + dummy.GetElement() );
const gdcmm::DictEntry &entry5 = dicts.GetDictEntry
    (dummy, dummy.GetOwner());
std::cout << entry5 << std::endl;

return 0;
}
```

27.102 ReadAndDumpDICOMDIR.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * This example shows how to read and dump a DICOMDIR File
 *
 * Thanks:
 * Tom Marynowski (lordglub gmail) for contributing this example
 */
#include "gdcmmReader.h"
#include "gdcmmMediaStorage.h"

typedef std::set<gdcmm::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];

    gdcmm::Reader reader;
    reader.SetFileName( filename);
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }
    std::stringstream strm;

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    gdcmm::FileMetaInformation &fmi = file.GetHeader
        ();

    gdcmm::MediaStorage ms;
    ms.SetFromFile(file);
    if( ms != gdcmm::MediaStorage::MediaStorageDirectoryStorage
        )
    {
        std::cout << "This file is not a DICOMDIR" << std::endl;
        return 1;
    }

    if (fmi.FindDataElement( gdcmm::Tag (0x0002, 0x0002)))
    {
        strm.str("");
        fmi.GetDataElement( gdcmm::Tag (0x0002, 0x0002) ).
            GetValue().Print(strm);
    }
    else
    {
        std::cerr << " Media Storage Sop Class UID not present" << std::cout;
    }
}
```

```

//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(
gdc::Tag (0x0008, 0x1030)))
            sqi->GetItem(itemused).GetDataElement(
gdc::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(
gdc::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement(
gdc::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(
gdc::Tag (0x0020, 0x000e)))
                sqi->GetItem(itemused).GetDataElement(
gdc::Tag (0x0020, 0x000e)).GetValue().Print(strm);
            std::cout << "          SERIE UID" << strm.str() << std::endl;

            //SERIE MODALITY
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(
gdc::Tag (0x0008, 0x0060)))
                sqi->GetItem(itemused).GetDataElement(
gdc::Tag (0x0008, 0x0060)).GetValue().Print(strm);
            std::cout << "          SERIE MODALITY" << strm.str() <<
std::endl;

            //SERIE DESCRIPTION
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(
gdc::Tag (0x0008, 0x103e)))
                sqi->GetItem(itemused).GetDataElement(
gdc::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(
gdc::Tag (0x0004, 0x1430)))
                sqi->GetItem(itemused).GetDataElement(
gdc::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(
(gdc::Tag (0x0004, 0x1511)))
                    sqi->GetItem(itemused).GetDataElement(
gdc::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(
(gdc::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;
}

```

27.103 ReadAndDumpDICOMDIR.py

```

#####
#
# 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.
#
# File: ReadAndDumpDICOMDIR.py
#
# Author: Lukas Batteau (lbatteau gmail)
#
# This example shows how to read and dump a DICOMDIR File.
# Based on Tom Marynowski's (lordglub gmail) example.
#
# Usage:
# python ReadAndDumpDICOMDIR.py [DICOMDIR file]
#####

import sys
import gdcmm

if __name__ == "__main__":
    # Check arguments
    if (len(sys.argv) < 2):
        # No filename passed
        print "No input filename found"
        quit()

    filename = sys.argv[1]

    # Read file
    reader = gdcmm.Reader()
    reader.SetFileName(filename)
    if (not reader.Read()):
        print "Unable to read %s" % (filename)
        quit()

```

```

file = reader.GetFile()

# Retrieve header information
fileMetaInformation = file.GetHeader()
print fileMetaInformation

# Retrieve data set
dataSet = file.GetDataSet()
#print dataSet

# Check media storage
mediaStorage = gdcm.MediaStorage()
mediaStorage.SetFromFile(file)
if (gdcm.MediaStorage.GetMSType(str(mediaStorage
)) != gdcm.MediaStorage.MediaStorageDirectoryStorage):
    # File is not a DICOMDIR
    print "This file is not a DICOMDIR (Media storage type: %s)" % (str(
mediaStorage))
    quit()

# Check Media Storage SOP Class
if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))) :
    sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0
x0002, 0x0002)).GetValue())
    # Check SOP UID
    if (sopClassUid != "1.2.840.10008.1.3.10"):
        # File is not a DICOMDIR
        print "This file is not a DICOMDIR"
else:
    # Not present
    print "Media Storage SOP Class not present"
    quit()

# Iterate through the DICOMDIR data set
iterator = dataSet.GetDES().begin()
while (not iterator.equal(dataSet.GetDES().end())):
    dataElement = iterator.next()

    # Check the element tag
    if (dataElement.GetTag() == gdcm.Tag(0x004, 0x1220)):
        # The 'Directory Record Sequence' element
        sequence = dataElement.GetValueAsSQ()

        # Loop through the sequence items
        itemNr = 1
        while (itemNr < sequence.GetNumberOfItems()):
            item = sequence.GetItem(itemNr)

            # Check the element tag
            if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))) :
                # The 'Directory Record Type' element
                value = str(item.GetDataElement(gdcm.Tag(0x0004, 0
x1430)).GetValue())

                # PATIENT
                while (value.strip() == "PATIENT"):
                    print value.strip()
                    # Print patient name
                    if (item.FindDataElement(gdcm.Tag(0x0010, 0
x0010))) :
                        value = str(item.GetDataElement(gdcm.Tag(0
x0010, 0x0010)).GetValue())
                        print value

                    # Print patient ID
                    if (item.FindDataElement(gdcm.Tag(0x0010, 0
x0020))) :
                        value = str(item.GetDataElement(gdcm.Tag(0
x0010, 0x0020)).GetValue())
                        print value

                    # Next
                    itemNr = itemNr + 1
                    item = sequence.GetItem(itemNr)
                    if (item.FindDataElement(gdcm.Tag(0x0004, 0
x1430))) :
                        value = str(item.GetDataElement(gdcm.Tag(0
x0004, 0x1430)).GetValue())

                # STUDY
                while (value.strip() == "STUDY"):

```

```

        print value.strip()

        # Print study UID
        if (item.FindDataElement(gdcm.Tag(0x0020, 0
x000d))) :
            value = str(item.GetDataElement(gdcm.Tag
(0x0020, 0x000d)).GetValue())
            print value

        # Print study date
        if (item.FindDataElement(gdcm.Tag(0x0008, 0
x0020))) :
            value = str(item.GetDataElement(gdcm.Tag
(0x0008, 0x0020)).GetValue())
            print value

        # Print study description
        if (item.FindDataElement(gdcm.Tag(0x0008, 0
x1030))) :
            value = str(item.GetDataElement(gdcm.Tag
(0x0008, 0x1030)).GetValue())
            print value

        # Next
        itemNr = itemNr + 1
        item = sequence.GetItem(itemNr)
        if (item.FindDataElement(gdcm.Tag(0x0004, 0
x1430))) :
            value = str(item.GetDataElement(gdcm.Tag
(0x0004, 0x1430)).GetValue())

            # SERIES
            while (value.strip() == "SERIES"):
                print value.strip()

            # Print series UID
            if (item.FindDataElement(gdcm.Tag(0
x0020, 0x000e))) :
                value = str(item.GetDataElement(gdcm.Tag
(0x0020, 0x000e)).GetValue())
                print value

            # Print series modality
            if (item.FindDataElement(gdcm.Tag(0
x0008, 0x0060))) :
                value = str(item.GetDataElement(gdcm.Tag
(0x0008, 0x0060)).GetValue())
                print "Modality"
                print value

            # Print series description
            if (item.FindDataElement(gdcm.Tag(0
x0008, 0x103e))) :
                value = str(item.GetDataElement(gdcm.Tag
(0x0008, 0x103e)).GetValue())
                print "Description"
                print value

            # Next
            itemNr = itemNr + 1
            item = sequence.GetItem(itemNr)
            if (item.FindDataElement(gdcm.Tag(0
x0004, 0x1430))) :
                value = str(item.GetDataElement(gdcm.Tag
(0x0004, 0x1430)).GetValue())

                # IMAGE
                while (value.strip() == "IMAGE"):
                    print value.strip()

                # Print image UID
                if (item.FindDataElement(gdcm.Tag(0
x0004, 0x1511))) :
                    value = str(item.GetDataElement(
gdcm.Tag(0x0004, 0x1511)).GetValue())
                    print value

            # Next
            if (itemNr < sequence.GetNumberOfItems()):
                itemNr = itemNr + 1
            else:

```

```

                break

                item = sequence.GetItem(itemNr)
                if (item.FindDataElement(gdcm.Tag(0
x0004, 0x1430))) :
                    value = str(item.GetDataElement(
gdcm.Tag(0x0004, 0x1430)).GetValue())

                # Next
                itemNr = itemNr + 1

```

27.104 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(0x0,0x0);
    //const DictEntry &del =
    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;
}

```

27.105 ReadExplicitLengthSQIVR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmImplicitDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmByteValue.h"
#include "gdcmSequenceOfItems.h"

using namespace gdcm;

int main(int argc, char *argv[])

```

```

{
    const char *filename = argv[1];
    gdcm::Reader r;
    r.SetFileName( filename );
    r.Read();

    //gdcm::PrivateTag pt(0x01,0x42,"ELSCINT1");
    //gdcm::Tag pt(0x88,0x200);
    gdcm::Tag pt(0x8,0x1140);
    DataSet &ds = r.GetFile().GetDataSet();
    const DataElement &de = ds.GetDataElement( pt );

    std::cout << de << std::endl;
    const ByteValue *bv = de.GetByteValue();
    SmartPointer<SequenceOfItems> sqi = new
        SequenceOfItems;
    sqi->SetLength( bv->GetLength() );
    std::stringstream ss;
    ss.str( std::string( bv->GetPointer(), bv->GetLength() ) )
        ;
    sqi->Read<ImplicitDataElement,SwapperNoOp>(
        ss );

    std::cout << *sqi << std::endl;

    return 0;
}

```

27.106 ReadFiles.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
import gdcm.*;
import java.io.File;

public class ReadFiles
{
    static int i = 0;
    public static void process(String path)
    {
        //String path = file.getPath();
        assert PosixEmulation.FileExists(path): "Problem converting to 8bits";

        System.out.println("Reading: " + path );
        System.out.println("File: " + i++);
        Reader r = new Reader();
        try
        {
            r.SetFileName( path );
            TagSetType skip = new TagSetType();
            skip.insert( new Tag(0x7fe0,0x10) );
            boolean b = r.ReadUpToTag( new Tag(0x88,0x200), skip );
            //System.out.println("DS:\n" + r.GetFile().GetDataSet().toString() );
        }
        finally
        {
            r.delete(); // will properly call C++ destructor and close file
            descriptor
        }
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())

```

```

        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
    else
    {
        process(dir.getPath());
    }
}

public static void waiting (int n)
{
    long t0, t1;
    t0 = System.currentTimeMillis();
    do
    {
        t1 = System.currentTimeMillis();
    }
    while ((t1 - t0) < (n * 1000));
}

public static void main(String[] args) throws Exception
{
    String directory = args[0];

    Directory gdir = new Directory();
    long n = gdir.Load( directory, true );
    System.out.println( gdir.toString() );
    FilenamesType files = gdir.GetFilenames();
    for( long i = 0; i < n; ++i )
    {
        String path = files.get( (int)i );
        process( path );
    }

    System.out.println( "Java API" );

    //waiting( 10 );
    for( int i = 0; i < 2; ++i )
    {
        File dir = new File(directory);
        visitAllFiles(dir);
    }
}

```

27.107 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 *name) 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;
}

```

27.108 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;
}

```

27.109 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 vtkgdcmm.*;
import vtk.*;

/*
 * Usage:
 * export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
 * java -classpath 'pwd'/vtkgdcmm.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("vtkgdcmmJava");
        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("MVSViewTKViewer.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() );
    }
}

```

27.110 ReadUTF8QtDir.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * GDCM API expect a const char * as input for SetFileName
 * In order to use this API from Qt, here is a simple test that
 * shows how to do it in a portable manner:
 *
 * http://doc.qt.nokia.com/latest/qdir.html#navigation-and-directory-operations
 */

#include "gdcmReader.h"
#include "gdcmDirectory.h"

#include <QDir>
#include <QString>
#include <QCoreApplication>

#include <string>
#include <fstream>

#include <stdio.h> // fopen

static int TestBothFuncs(const char *info , const char *ba_str)
{
    int res = 0;
    FILE *f = fopen( ba_str, "r" );
    if( f )
    {
        std::cout << info << " fopen: " << ba_str << std::endl;
        fclose(f);
        ++res;
    }
    gdcm::Reader reader;
    std::ifstream is( ba_str );
    if( is.is_open() )
    {
        std::cout << info << " is_open: " << ba_str << std::endl;
        ++res;
    }
    reader.SetStream( is );
    if( reader.CanRead() == true )
    {
        std::cout << info << " SetStream/CanRead:" << ba_str << std::endl;
        ++res;
    }
    is.close();
    reader.SetFileName( ba_str );
    if( reader.CanRead() == true )
    {
        std::cout << info << " SetFileName/CanRead:" << ba_str << std::endl;
        ++res;
    }
    return 4 - res;
}

static int scanFolder(const char dirname[])
{
    int res = 0;
    gdcm::Directory dir;
    unsigned int nfiles = dir.Load( dirname, true );
    const gdcm::Directory::FileNamesType &filenames
        = dir.GetFilesNames();

    for( unsigned int i = 0; i < nfiles; ++i )
    {
        const char *ba_str = filenames[i].c_str();
        res += TestBothFuncs("GDCM",ba_str);
    }
}

```

```

    }
    return res;
}

static int scanFolderQt(QDir const &dir, QStringList& files)
{
    int res = 0;
    QFileInfoList children = dir.entryInfoList(QDir::AllEntries|
        QDir::NoDotAndDotDot);
    for ( int i=0; i<children.count(); i++ ) {
        QFileInfo file = children.at(i);
        if ( file.isDir() == true ) {
            res += scanFolderQt(QDir(file.absoluteFilePath()), files);
            continue;
        }
        // Convert back from the internal representation to 8bits
        // toLocal8Bit() returns by copy. Need to store explicitly the QByteArray
        QByteArray str = file.absoluteFilePath().toLocal8Bit();
        const char *ba_str1 = str.constData();
        res += TestBothFuncs("QString", ba_str1);
    }
    return res;
}

int main(int argc, char *argv[])
{
    // very important:
    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;
}

```

27.111 RefCounting.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using vtkgdcml;

/*
 * this is not so much an example but simply a test to make sure cstor / dstor
 * work as expected
 * and call the ::New and ->Delete() of VTK style.
 */
public class RefCounting
{
    public static int Main(string[] args)

```

```

{
    vtkGDCMTesting testing1 = vtkGDCMTesting.New
    ();
    vtkGDCMTesting testing2 = new vtkGDCMTesting();
    // just in case people do not read STYLE documentation

    vtkGDCMImageReader reader1 = vtkGDCMImageReader
    .New();
    vtkGDCMImageReader reader2 = new vtkGDCMImageReader
    ();

    vtkGDCMImageWriter writer1 = vtkGDCMImageWriter
    .New();
    vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter
    ();

    using (vtkGDCMTesting testing3 = new vtkGDCMTesting
    ())
    {
        System.Console.Write( "GetReferenceCount: " + testing1.GetReferenceCount(
        ) + "\n");
        System.Console.Write( "GetReferenceCount: " + testing2.GetReferenceCount(
        ) + "\n");
        System.Console.Write( "GetReferenceCount: " + testing3.GetReferenceCount(
        ) + "\n");
    }

    using (vtkGDCMImageReader reader3 = new
    vtkGDCMImageReader())
    {
        System.Console.Write( "GetReferenceCount: " + reader3.GetReferenceCount(
        ) + "\n");
    }

    using (vtkGDCMImageWriter writer3 = vtkGDCMImageWriter
    .New())
    {
        System.Console.Write( "GetReferenceCount: " + writer3.GetReferenceCount(
        ) + "\n");
    }

    // C# destructor will call ->Delete on all C++ object as expected.
    return 0;
}
}

```

27.112 ReformatFile.cs

This is a C++ example on how to use `gdcm::FileDerivation`

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ReformatFile.exe input.dcm output.dcm
 */
using System;
using gdcm;

public class ReformatFile
{

```

```

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Reformat App"
    );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot()
    );

    string filename = args[0];
    string outfilename = args[1];

    Reader reader = new Reader();
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        System.Console.WriteLine( "Could not read: " + filename );
        return 1;
    }

    UIDGenerator uid = new UIDGenerator(); // helper for uid generation
    FileDerivation fd = new FileDerivation();
    // For the pupose of this excise 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;
}

```

27.113 RemovePrivateTags.py

```

#####
#
# Program: GDCM (Grassroots DICOM). A DICOM library
#
# Copyright (c) 2006-2011 Mathieu Malaterre
# All rights reserved.
# See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

```
#
# This software is distributed 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:

python RemovePrivateTags.py input.dcm output.dcm
"""

import sys
import gdcm

if __name__ == "__main__":

    file1 = sys.argv[1]
    file2 = sys.argv[2]

    # Instanciate the reader.
    r = gdcm.Reader()
    r.SetFileName( file1 )
    if not r.Read():
        sys.exit(1)

    # Remove private tags
    ano = gdcm.Anonymizer()
    ano.SetFile( r.GetFile() )
    if not ano.RemovePrivateTags():
        sys.exit(1)

    # Write DICOM file
    w = gdcm.Writer()
    w.SetFile( ano.GetFile() )
    #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
    w.SetFileName( file2 )
    if not w.Write():
        sys.exit(1)

    # It is usually a good idea to exit the script with an error, as gdcm does
    # not remove partial (incorrect) DICOM file
    # (application level)
```

27.114 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;
}
}

```

27.115 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.GetFilesNames();

    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.GetFilesNames();
    vtkStringArray *vtkfiles = vtkStringArray::New();
    std::vector< std::string >::const_iterator it = sorted.begin();
    for( ; it != sorted.end(); ++it)
    {
        const std::string &f = *it;
        vtkfiles->InsertNextValue( f.c_str() );
    }

    //_reader->SetDirectoryName(fileName);
    //_reader->SetFileNames( d->GetFiles() );
    _reader->SetFileNames( vtkfiles );
    _reader->Update();

    const vtkFloatingPointType *spacing = _reader->GetOutput()->GetSpacing();

```

```

vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
v16->SetInput( _reader->GetOutput() );
v16->SetOutputSpacing( spacing[0], spacing[1], ippszspacing );
v16->Update();

_threshold=vtkImageThreshold::New();
_threshold->ThresholdByUpper(-3024.0);
_threshold->ReplaceOutOn();
_threshold->SetOutValue(0.0);
_threshold->SetInputConnection(v16->GetOutputPort());

_shift=vtkImageShiftScale::New();
_shift->SetShift(0);
_shift->SetScale(1);
_shift->SetInputConnection(_threshold->GetOutputPort());

// Initialize the reslice with an axial orientation.
vtkSmartPointer<vtkMatrix4x4> matrix =
    vtkSmartPointer<vtkMatrix4x4>::New();
matrix->Identity();

_transform = vtkTransform::New();
_transform->SetMatrix(matrix);

_reslice = vtkImageReslice::New();
_reslice->SetOutputDimensionality(3);

// PROBLEM:
// The original intent was to connect the same transform
// to the vtkImageReslice and vtkTransformPolyDataFilter,
// but the resulting reslices appear different using the
// vtkTransform as opposed to explicitly setting the
// reslice axes via SetResliceAxes. Also, if the vtkTransform
// is connected and orientated other than axial, the extents
// don't seem to update resulting in VTK believing the slice
// is out of range.

_reslice->SetResliceTransform(_transform);
_reslice->SetResliceAxes(matrix);
_reslice->SetInputConnection(_reader->GetOutputPort());
_reslice->SetInputConnection(_shift->GetOutputPort());

// Create the sphere target shape.
_sphere=vtkSphereSource::New();
_sphere->SetRadius(7.0);
_sphere->SetThetaResolution(16);
_sphere->SetPhiResolution(16);
_sphere->SetCenter(sphereCenter[0], sphereCenter[1], sphereCenter[2]);

_sphereMapper=vtkPolyDataMapper::New();
_sphereMapper->SetInputConnection(_sphere->GetOutputPort());

_sphereActor=vtkActor::New();
_sphereActor->SetMapper(_sphereMapper);
_sphereActor->PickableOff();
_sphereActor->GetProperty()->SetColor(1.0, 0.0, 0.0);
_sphereActor->GetProperty()->SetEdgeColor(1.0, 0.0, 0.0);
_sphereActor->GetProperty()->SetDiffuseColor(1.0, 0.0, 0.0);
_sphereActor->SetVisibility(true);

// Create the cutting pipeline.
// This plane will be positioned in the original image coordinate
// system.
_plane = vtkPlane::New();
_plane->SetNormal(0.0, 0.0, 1.0);

_cutter = vtkCutter::New();
_cutter->SetInputConnection(_sphere->GetOutputPort());
_cutter->SetCutFunction(_plane);
_cutter->GenerateCutScalarsOn();
_cutter->SetValue(0, 0.5);

// The transform attached to _polyTransform should move the cut
// ROI into the resliced coordinate system, which should be the
// same as the coordinate system of the resliced images.
// PROBLEM: It doesn't.
_polyTransform = vtkTransformPolyDataFilter::New();
_polyTransform->SetTransform(_transform);
_polyTransform->SetInputConnection(_cutter->GetOutputPort());

```

```

        _ROIMapper = vtkPolyDataMapper2D::New();
        _ROIMapper->SetInputConnection(_polyTransform->GetOutputPort());

vtkCoordinate* coordinate = vtkCoordinate::New();
coordinate->SetCoordinateSystemToWorld();
_ROIMapper->SetTransformCoordinate(coordinate);

_ROIActor = vtkActor2D::New();
_ROIActor->SetMapper(_ROIMapper);

// Make sure the cut can be seen, especially the edges.
props=_ROIActor->GetProperty();
props->SetLineWidth(2);
props->SetOpacity(1.0);
// props->EdgeVisibilityOn();
// props->SetDiffuse(0.8);
// props->SetSpecular(0.3);
// props->SetSpecularPower(20);
// props->SetRepresentationToSurface();
// props->SetDiffuseColor(1.0, 0.0, 0.0);
// props->SetEdgeColor(1.0, 0.0, 0.0);
props->SetColor(1.0, 0.0, 0.0);

_interactor = vtkRenderWindowInteractor::New();

// Create the image viewer and add the actor with the cut ROI.
_imageViewer = vtkImageViewer2::New();
_imageViewer->SetupInteractor(_interactor);
_imageViewer->SetSize(400, 400);
_imageViewer->SetColorWindow(1024);
_imageViewer->SetColorLevel(800);
_imageViewer->SetInputConnection(_reslice->GetOutputPort());
_imageViewer->GetImageActor()->SetOpacity(0.5);

_annotation = vtkTextActor::New();
_annotation->SetTextScaleModeToViewport();
_imageViewer->GetRenderer()->AddActor(_annotation);

// Add the cut shape actor to the renderer.
_imageViewer->GetRenderer()->AddActor(_ROIActor);

// Set up the key handler.
vtkSmartPointer<KeyCallback> callback =
vtkSmartPointer<KeyCallback>::New();
callback->SetCallbackData(this);
_interactor->AddObserver(vtkCommand::KeyPressEvent, callback);

_interactor->Initialize();
}

void Start()
{
    _interactor->Start();
}

void ResetOrientation()
{
    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();
    matrix->Identity();

    SetOrientation(matrix);
}

// Make sure the orientation of the vtkImageReslice and
// vtkTransform are in sync.
void SetOrientation(vtkMatrix4x4* matrix)
{
    _reslice->SetResliceAxes(matrix);
    _reslice->Update();

    vtkMatrix4x4* inverse = vtkMatrix4x4::New();
    vtkMatrix4x4::Invert(matrix, inverse);

    _transform->SetMatrix(inverse);
    _transform->Update();
}

// Set the current slice of the current view.
void SetSlice(int slice)

```

```

{
    std::stringstream posString;

    double    center[3];
    double    spacing[3];
    double    origin[3];
    double    point[4];
    double    newPoint[4];

    vtkImageData* imageData;
    int newSlice;

    // Try to make sure the extents of the reslice are updated.
    // PROBLEM: It doesn't seem to work when changing the orientation.
    imageData=vtkImageData::SafeDownCast(_reslice->GetOutput());
    imageData->UpdateInformation();

    // Let vtkImageViewer2 handle the slice limits.
    _imageView->SetSlice(slice);
    newSlice=GetSlice();

    imageData->GetCenter(center);
    imageData->GetSpacing(spacing);
    imageData->GetOrigin(origin);

    // Compute the position of the center of the slice based on the
    // spacing of the slices. The resliced axis will always
    // be the "Z" axis.
    point[0]=center[0];
    point[1]=center[1];
    point[2]=(newSlice * spacing[2]) + origin[2];
    point[3]=1.0;

    // Convert the coordinate from the reslice coordinate system to the
    // original image coordinate system.
    // PROBLEM: Logically this seems like it should have been multiplied
    // by the inverse to translate from the resliced coordinate system to
    // the original coordinate system. However, multiplying by the inverse
    // sticks the plane in the wrong place completely. Using the original
    // matrix at least gets the Z coordinate right.
    vtkMatrix4x4* matrix=_reslice->GetResliceAxes();
    vtkSmartPointer<vtkMatrix4x4> inverse =
        vtkSmartPointer<vtkMatrix4x4>::New();
    vtkMatrix4x4::Invert(matrix, inverse);

    matrix->MultiplyPoint(point, newPoint);
    _plane->SetOrigin(newPoint[0], newPoint[1], newPoint[2]);

    // Annotate the image.
    posString << "Position: (" << newPoint[0] << ", " << newPoint[1]
        << ", " << newPoint[2] << ") Slice: " << newSlice;
    _annotation->SetInput(posString.str());

    _imageView->Render();
}

int GetSlice()
{
    return _imageView->GetSlice();
}

// Set the orientation of the view.
void SetOrientation(ResliceRenderer::ORIENTATION orientation)
{
    vtkCamera* camera=_imageView->GetRenderrer()->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());
imageData->UpdateInformation();
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.
_imageViewer->SetInput(NULL);
_imageViewer->SetInputConnection(_reslice->GetOutputPort());

_imageViewer->GetRenderer()->ResetCameraClippingRange();
_imageViewer->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(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)
{
    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;
}

```

27.116 ReWriteSCAsMR.py

```

#####
#
# Program: GDCM (Grassroots DICOM). A DICOM library
#
# Copyright (c) 2006-2011 Mathieu Malaterre
# All rights reserved.
# See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
# This software is distributed 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 1.x would write out MR Image Storage as Secondary Capture Object while
still setting Rescale Slope/Intercept
and saving the Pixel Spacing in (0028,0030)
"""

import gdcm
import sys,os

def CheckSecondaryCaptureObjectIsMRImageStorage(r):
    ds = r.GetFile().GetDataSet()
    # Check Source Image Sequence
    if ds.FindDataElement( gdcm.Tag(0x0008,0x2112) ):
        sis = ds.GetDataElement( gdcm.Tag(0x0008,0x2112) )
        sqsis = sis.GetSequenceOfItems()
        if sqsis.GetNumberOfItems():
            item1 = sqsis.GetItem(1)
            nestedds = item1.GetNestedDataSet()
            if nestedds.FindDataElement( gdcm.Tag(0x0008,0x1150) ):
                ReferencedSOPClassUID = nestedds.GetDataElement( gdcm.Tag(0
                x0008,0x1150) )
                raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
                uids = gdcm.UIDs()
                # what is the actual object we are looking at ?
                ms = gdcm.MediaStorage()
                ms.SetFromDataSet(ds)
                msuid = ms.GetString()
                uids.SetFromUID( msuid )
                msuidname = uids.GetName() # real Media Storage Name
                uids.SetFromUID( raw )
                sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
                # If object is SC and Source derivation is MRImageStorage then we can
                assume 'Pixel Spacing' is correct
                if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary
                Capture Image Storage' ):
                    return True
            # in all other case simply return the currentspacing:
            return False

if __name__ == "__main__":
    r = gdcm.ImageReader()
    filename = sys.argv[1]
    r.SetFileName( filename )
    if not r.Read():
        sys.exit(1)
    f = r.GetFile()

```

```

if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
    # Special handling of the spacing:
    # GDCM 1.2.0 would not rewrite correctly DICOM Object and would always set
    # them as 'Secondary Capture Image Storage'
    # while we would rather have 'MR Image Storage'
    gdcml.ImageHelper.SetForcePixelSpacing(
        True )
    mrspacing = gdcml.ImageHelper.GetSpacingValue
        ( r.GetFile() )
    # TODO: I cannot do simply the following:
    #image.SetSpacing( mrspacing )
    image.SetSpacing(0, mrspacing[0] )
    image.SetSpacing(1, mrspacing[1] )
    image.SetSpacing(2, mrspacing[2] )
    gdcml.ImageHelper.SetForceRescaleInterceptSlope
        ( True )
    ris = gdcml.ImageHelper.GetRescaleInterceptSlopeValue
        ( r.GetFile() )
    image.SetIntercept( ris[0] )
    image.SetSlope( ris[1] )

outfilename = sys.argv[2]
w = gdcml.ImageWriter()
w.SetFileName( outfile )
w.SetFile( r.GetFile() )
w.SetImage( image )
if not w.Write():
    sys.exit(1)

sys.exit(0)

```

27.117 rle2img.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to rewrite a ELSCINT1/PMSCT_RLE1 compressed
 * image so that it is readable by most 3rd party software (DICOM does
 * not specify this particular encoding).
 * This is required for the sake of interoperability with any standard
 * conforming DICOM system.
 *
 * Everything done in this code is for the sole purpose of writing
 * interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcml-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Mauro Maiorca for bringing to our attention on this new ELSCINT1
 * compression algorithm : PMSCT_RLE1 (different from the 'LOSSLESS RICE')
 * See post at:
 * http://groups.google.com/group/comp.protocols.dicom/msg/f2b99bf706a7f8ca
 *
 * Thanks to Jesus Spinola, for more datasets,
 * http://www.itk.org/pipermail/insight-users/2008-April/025571.html
 *
 * And last but not least, a very big thank to Ivo van Poorten, without
 * whom we would still be looking at this compressed byte stream as if
 * it was RLE compressed.
 */

```

```

#include "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];
            int value = v2 * 256 + v1;
            output.push_back( value );
            delta = value;
            i+=2;
        }
        else
        {
            int value = 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 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet
        ();

    // (07a1,1011) CS [PMSCT_RLE1] # 10,1
    // Tamar Compression Type
    const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"
        ELSCINT1");
    if( !ds.FindDataElement( tcompressiontype ) ) return 1;
    const gdcm::DataElement& compressiontype = ds.GetDataElement

```

```

        ( tcompressiontype );
    if ( compressiontype.IsEmpty() ) return 1;
    const gdcm::ByteValue * bv = compressiontype.GetByteValue
        ();
    std::string comprle = "PMSCT_RLE1";
    std::string comprgb = "PMSCT_RGB1";
    bool isrle = false;
    bool isrgb = false;
    if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) ==
        0 )
    {
        isrle = true;
    }
    if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) ==
        0 )
    {
        isrgb = true;
        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();

    std::vector<unsigned short> buffer;
    delta_decode(bv2->GetPointer(), bv2->GetLength(), buffer);

    gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0
        x0010) );
    pixeldata.SetVR( gdcm::VR::OW );
    pixeldata.SetByteValue( (char*)&buffer[0], 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 = "outrle.dcm";
    writer.SetFileName( outfilename.c_str() );
    if( !writer.Write() )
    {
        std::cerr << "Failed to write" << std::endl;
        return 1;
    }

    std::cout << "success !" << std::endl;

    return 0;
}

```

27.118 rtstructapp.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMPolyDataReader.h"
#include "vtkGDCMPolyDataWriter.h"

#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkAppendPolyData.h"
#include "vtkImageData.h"

/*
 * Small example to read in a RTSTRUCT and write it out (displays it too).
 */

// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];
    vtkGDCMPolyDataReader * reader =
        vtkGDCMPolyDataReader::New();
    reader->SetFileName( filename );
    reader->Update();

    //std::cout << reader->GetMedicalImageProperties()->GetStudyDate() <<
        std::endl;

    vtkGDCMPolyDataWriter * writer =
        vtkGDCMPolyDataWriter::New();
    writer->SetNumberOfInputPorts( reader->
        GetNumberOfOutputPorts() );
    writer->SetFileName( outfilename );
    for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
        writer->SetInput( num, reader->GetOutput(num) );
    //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)

```

```

    {
        append->AddInput( reader->GetOutput(i) );
    }

    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    cubeMapper->SetInput( append->GetOutput() );
    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;
}

```

27.119 ScanDirectory.cs

This is a C# example on how to use `gdcm::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:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ScanDirectory.exe /path/to/gdcmData/
 */
using System;
using gdcm;

public class ScanDirectory
{
    public static int Main(string[] args)
    {
        string directory = args[0];
        Tag t = new Tag(0x8,0x8);

        Directory d = new Directory();
        uint nfiles = d.Load( directory );
    }
}

```

```

    if(nfiles == 0) return 1;
    //System.Console.WriteLine( "Files:\n" + d.toString() );

    //Scanner s = new Scanner();
    SmartPtrScan sscan = Scanner.New();
    Scanner s = sscan.__ref__();
    SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
    s.AddTag( t );
    bool b = s.Scan( d.GetFileNames() );
    if(!b) return 1;

    System.Console.WriteLine( "Scan:\n" + s.toString() );

    System.Console.WriteLine( "success" );
    return 0;
}
}

```

27.120 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" );
}

```

27.121 ScanDirectory.py

```

#####
#

```

```

# Program: GDCM (Grassroots DICOM). A DICOM library
#
# Copyright (c) 2006-2011 Mathieu Malaterre
# All rights reserved.
# See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
# This software is distributed 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 sys,os

if __name__ == "__main__":
    directory = sys.argv[1]

    # Define the set of tags we are interested in
    t1 = gdcm.Tag(0x8,0x8);
    t2 = gdcm.Tag(0x10,0x10);

    # Iterate over directory
    d = gdcm.Directory();
    nfiles = d.Load( directory );
    if(nfiles == 0): sys.exit(1);
    # System.Console.WriteLine( "Files:\n" + d.toString() );

    filenames = d.GetFilenames()

    # Get rid of any Warning while parsing the DICOM files
    gdcm.Trace.WarningOff()

    # instanciate Scanner:
    s = gdcm.Scanner();
    s.AddTag( t1 );
    s.AddTag( t2 );
    b = s.Scan( filenames );
    if(not b): sys.exit(1);

    print "success" ;
    #print s

    pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
    pttv.Start()
    # iterate until the end:
    while( not pttv.IsAtEnd() ):
        # get current value for tag and associated value:
        # if tag was not found, then it was simply not added to the internal
        # std::map
        # Warning value can be None
        tag = pttv.GetCurrentTag()
        value = pttv.GetCurrentValue()
        print tag,"->",value
        # increment iterator
        pttv.Next()

    sys.exit(0)

```

27.122 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;
    }
}

```

27.123 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;
}
}

```

27.124 SimplePrintPatientName.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrintPatientName.exe gdcmData/012345.002.050.dcm
 */
/*
This example was provided by Jonathan Morra /jonmorra gmail com/
on the gdcm mailing list (Fri, 28 May 2010)
*/
using System;
using gdcm;

namespace GDCMTest
{
    class SimplePrintPatientName
    {
        static int Main(string[] args)
        {
            if (args.Length != 1)
            {
                Console.WriteLine("This program prints the patient name of a dicom file with gdcm");
                Console.WriteLine("Usage: [input.dcm]");
                return 1;
            }
        }
    }
}

```

```

gdcM.Reader reader = new gdcM.Reader();
reader.SetFileName(args[0]);
bool ret = reader.Read();
//TagSetType tst = new TagSetType();
//tst.Add( new Tag(0x7fe0,0x10) );
//bool ret = reader.ReadUpToTag( new Tag(0x88,0x200), tst );
if( !ret )
{
    return 1;
}

gdcM.File file = reader.GetFile();

gdcM.StringFilter filter = new gdcM.StringFilter();
filter.SetFile(file);
string value = filter.ToString(new gdcM.Tag(0x0010, 0x0010));

Console.WriteLine("Patient Name: " + value);
return 0;
}
}

```

27.125 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 "gdcMScanner.h"

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::Scanner s;

    const gdcM::Tag tag_array[] = {
        gdcM::Tag(0x8,0x50),
        gdcM::Tag(0x8,0x51),
        gdcM::Tag(0x8,0x60),
    };
    s.AddTag( tag_array[0] );
    s.AddTag( tag_array[1] );

```

```

s.AddTag( tag_array[2] );

gdcmm::Directory::FileNamesType filenames;
filenames.push_back( filename );
filenames.push_back( filename_invalid );

if( !s.Scan( filenames ) )
{
    return 1;
}

//s.Print( std::cout );

if( s.IsKey( filename ) )
{
    std::cout << "INFO:" << filename << " is a proper Key for the Scanner (this
        is a DICOM file)" << std::endl;
}

if( !s.IsKey( filename_invalid ) )
{
    std::cout << "INFO:" << filename_invalid << " is not a proper Key for the
        Scanner (this is either not a DICOM file or file does not exist)" << std::endl;
}

gdcmm::Scanner::TagToValue const &ttv = s.GetMapping
    (filename);

const gdcmm::Tag *ptag = tag_array;
for( ; ptag != tag_array + 3; ++ptag )
{
    gdcmm::Scanner::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;
}

```

27.126 SortImage.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 "gdcmSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

bool mysort(gdcm::DataSet const & ds1, gdcm::DataSet
            const & ds2 )
{
    //gdcm::Attribute<0x0020,0x0013> at1; // Instance Number
    gdcm::Attribute<0x0018,0x1060> at1; // Trigger
        Time
    gdcm::Attribute<0x0020,0x0032> at11; // Image
        Position (Patient)
    at1.Set( ds1 );
    at11.Set( ds1 );
    //gdcm::Attribute<0x0020,0x0013> at2;
    gdcm::Attribute<0x0018,0x1060> at2;
    gdcm::Attribute<0x0020,0x0032> at22;
    at2.Set( ds2 );
    at22.Set( ds2 );
    if( at11 == at22 )
    {
        return at1 < at2;
    }
    return at11 < at22;
}

bool mysort_part1(gdcm::DataSet const & ds1, gdcm::DataSet
                  const & ds2 )
{
    gdcm::Attribute<0x0018,0x1060> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0018,0x1060> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort_part2(gdcm::DataSet const & ds1, gdcm::DataSet
                  const & ds2 )
{
    gdcm::Attribute<0x0020,0x0032> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x0032> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

// technically all files are in the same Frame of Reference, so this function
// should be a no-op
bool mysort_dummy(gdcm::DataSet const & ds1, gdcm::DataSet
                  const & ds2 )
{
    gdcm::Attribute<0x0020,0x0052> at1; //
        FrameOfReferenceUID
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x0052> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

int main(int argc, char *argv[])
{
    const char *dirname = argv[1];
    gdcm::Directory dir;
    unsigned int nfiles = dir.Load( dirname );

    dir.Print( std::cout );

    gdcm::Sorter sorter;
    sorter.SetSortFunction( mysort );
    sorter.Sort( dir.GetFileNames() );

    std::cout << "Sorter:" << std::endl;
    sorter.Print( std::cout );

    gdcm::Sorter sorter2;
    sorter2.SetSortFunction( mysort_part1 );
    sorter2.StableSort( dir.GetFileNames() );
}

```

```

    sorter2.SetSortFunction( mysort_part2 );
    sorter2.StableSort( sorter2.GetFileNames() ); //
        IMPORTANT
    sorter2.SetSortFunction( mysort_dummy );
    sorter2.StableSort( sorter2.GetFileNames() ); //
        IMPORTANT

    std::cout << "Sorter2:" << std::endl;
    sorter2.Print( std::cout );

    gdcm::Scanner s;
    s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( dir.GetFileNames() );

    //s.Print( std::cout );

    // Count how many different IPP there are:
    const gdcm::Scanner::ValueType &values = s.
        GetValues();
    unsigned int 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;
}

```

27.127 SortImage.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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:

    python SortImage.py dirname
"""

import gdcm
import sys

def PrintProgress(object, event):
    assert event == "ProgressEvent"
    print "Progress:", object.GetProgress()

def MySort(ds1, ds2):
    # compare ds1
    return False

if __name__ == "__main__":

    dirname = sys.argv[1]
    d = gdcm.Directory()
    d.Load( dirname )

    print d

```

```

sorter = gdcm.Sorter()
sorter.SetSortFunction( MySort )
#sorter.AddObserver( "ProgressEvent", PrintProgress )
sorter.Sort( d.GetFileNames() )

print "Sorter:"
print sorter

```

27.128 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;
    }
}

```

27.129 StandardizeFiles.cs

This is a C++ example on how to use `gdcm::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.Bitmap bitout = change.GetOutput();
        gdcm.Pixmap pixout = (gdcm.Pixmap)bitout;
        //System.Console.WriteLine( "Debug: " + pixout.toString() );

        writer.SetPixmap( pixout );
        if ( !writer.Write() )
        {
            System.Console.WriteLine( "Could not write: " + outfilename );
            return false;
        }

        return true;
    }

    public static int Main(string[] args)
    {
        gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Standardize
            App" );

        // http://www.oid-info.com/get/1.3.6.1.4.17434
        string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
        gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
        System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot()
            );

        string dir1 = args[0];
        string dir2 = args[1];

        // Check input is valid:
        if ( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
        {
            System.Console.WriteLine( "Input directory: " + dir1 + " does not exist.
                Sorry" );
            return 1;
        }
        if ( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
        {
            System.Console.WriteLine( "Output directory: " + dir2 + " does not exist.
                Sorry" );
            return 1;
        }

        Directory d = new Directory();
    }
}

```

```

uint nfiles = d.Load( dir1, true );
if(nfiles == 0) return 1;

// Process all filenames:
FileNamesType filenames = d.GetFilenames();
for( uint i = 0; i < nfiles; ++i )
{
    string filename = filenames[ (int)i ];
    string outfilename = filename.Replace( dir1, dir2 );
    System.Console.WriteLine( "Filename: " + filename );
    System.Console.WriteLine( "Out Filename: " + outfilename );
    if( !ProcessOneFile( filename, outfilename ) )
    {
        System.Console.WriteLine( "Could not process filename: " + filename );
        //return 1;
    }
}

return 0;
}

```

27.130 StreamImageReaderTest.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmlStreamImageReader.h"
#include "gdcmlFileMetaInformation.h"
#include "gdcmlSystem.h"
#include "gdcmlFilename.h"
#include "gdcmlByteSwap.h"
#include "gdcmlTrace.h"
#include "gdcmlTesting.h"
#include "gdcmlImageHelper.h"
#include "gdcmlImageReader.h"
#include "gdcmlImage.h"
#include "gdcmlMediaStorage.h"
#include "gdcmlRAWCodec.h"
#include "gdcmlJPEGLSCodec.h"
#include "gdcmlUIDGenerator.h"
#include "gdcmlStreamImageWriter.h"
#include "gdcmlAttribute.h"
#include "gdcmlFile.h"
#include "gdcmlTag.h"

bool StreamImageRead(gdcml::StreamImageWriter &
    theStreamWriter,
    const char* filename, const char* outfilename, int resolution)
{
    gdcml::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 =
    gdcmm::ImageHelper::GetDimensionsValue(
        reader.GetFile());
// std::cout << extent[0];
//at this point, these values aren't used, but may be in the future
//unsigned short xmin = 0;
//unsigned short xmax = extent[0];
//unsigned short ymin = 0;
//unsigned short ymax = extent[1];
//unsigned short zmin = 0;
//unsigned short zmax = extent[2];

std::cout<< "\n Row: "<<extent[0] <<"\n Col :"<< extent[1]<< "\n Resolution :
    "<< extent[2] << std::endl;

int a =1;
for (int i=1; i<=(extent[2]-resolution);++i)
    a = a*2;

reader.DefinePixelExtent(0, extent[0]/a, 0, extent[1]/a,
    resolution-1, resolution);

unsigned long len = reader.DefineProperBufferLength()
    ;
char* finalBuffer = new char[len];
memset(finalBuffer, 0, sizeof(char)*len);

if (reader.CanReadImage())
{
    bool result = reader.Read(finalBuffer, len);
    if( !result )
    {
        std::cout << "res2 failure:" << filename << std::endl;
        delete [] finalBuffer;
        return 1;
    }
    else
    {
        std::cout<< "Able to read";
    }
}
else
{
    std::cerr<< "Not able to put in buffer"<< std::endl;
}
/*
//now, read in smaller buffer extents
reader.DefinePixelExtent(xmin, xmax, ymin, ymax);
len = reader.DefineProperBufferLength();

char* buffer = new char[len];
bool res2 = reader.Read(buffer, len);
if( !res2 ){
    std::cerr << "res2 failure:" << filename << std::endl;
    return 1;
}
//copy the result into finalBuffer
memcpy(finalBuffer, buffer, len);

//now read the next half of the image
ymin = ymax;
ymax = extent[1];

reader.DefinePixelExtent(xmin, xmax, ymin, ymax);

//std::cerr << "Success to read image from file: " << filename <<
    std::endl;
unsigned long len2 = reader.DefineProperBufferLength();

char* buffer2 = new char[len2];
bool res3 = reader.Read(buffer2, len2);
if( !res3 ){
    std::cerr << "res3 failure:" << filename << std::endl;
    return 1;
}
//copy the result into finalBuffer
memcpy(&(finalBuffer[len]), buffer2, len2);

delete [] buffer;
delete [] buffer2;

```

```

*/

gdcM::Writer w;
gdcM::File &file = w.GetFile();
gdcM::DataSet &ds = file.GetDataSet();

file.GetHeader().SetDataSetTransferSyntax(
    gdcM::TransferSyntax::ExplicitVRLittleEndian
);

gdcM::UIDGenerator uid;
gdcM::DataElement de( gdcM::Tag(0x8,0x18) ); // SOP
    Instance UID
de.SetVR( gdcM::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );

gdcM::DataElement del( gdcM::Tag(0x8,0x16) );
del.SetVR( gdcM::VR::UI );
gdcM::MediaStorage ms(
    gdcM::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()
));
ds.Insert( del );

const char mystr[] = "MONOCHROME2 ";
gdcM::DataElement de2( gdcM::Tag(0x28,0x04) );
//de.SetTag(gdcM::Tag(0x28,0x04));
de2.SetVR( gdcM::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcM::Attribute<0x0028,0x0008>
    Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0010> row = {extent[0
]/a};//
ds.Insert( row.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0011> col = {extent[1
]/a};//
ds.Insert( col.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0002> at1 = {1};//
ds.Insert( at1.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );
/*
ds1.Remove( gdcM::Tag(0x0028,0x0008) );

gdcM::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds1.Insert( Number_Of_Frames.GetAsDataElement() );
*/

theStreamWriter.SetFile(file);

if (!theStreamWriter.WriteImageInformation())
{
    std::cerr << "unable to write image information" << std::endl;
    return 1; //the CanWrite function should prevent getting here, else,
    //that's a test failure
}
std::vector<unsigned int> extent1 = gdcM::ImageHelper::GetDimensionsValue
    (file);

unsigned short xmax = extent1[0];
unsigned short ymax = extent1[1];
unsigned short theChunkSize = 1;
unsigned short ychunk = extent1[1]/theChunkSize; //go in chunk sizes of
    theChunkSize
unsigned short zmax = 1;

```

```

std::cout<< "\n Row: "<<extent1[0] <<"\n Col :"<< extent1[1]<< "\n
Resolution :"<< extent1[2] << std::endl;

if (xmax == 0 || ymax == 0)
{
std::cerr << "Image has no size, unable to write zero-sized image." <<
std::endl;
return 0;
}

int z, y, nexty;
unsigned long prevLen = 0; //when going through the char buffer, make sure
to grab
//the bytes sequentially. So, store how far you got in the buffer with
each iteration.

for (z = 0; z < zmax; ++z){
for (y = 0; y < ymax; y += ychunk){
nexty = y + ychunk;
if (nexty > ymax) nexty = ymax;
theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z
, z+1);
unsigned long len = theStreamWriter.DefineProperBufferLength
();
std::cout << "\n" <<len;
char* finalBuffer1 = new char[len];
memcpy(finalBuffer1, &(finalBuffer[prevLen]), len);
std::cout << "\nable to write";

if (!theStreamWriter.Write(finalBuffer1, len)){
std::cerr << "writing failure:" << "output.dcm" << " at y = " << y <<
" and z= " << z << std::endl;
delete [] finalBuffer1;
delete [] finalBuffer;
return 1;
}
delete [] finalBuffer1;
prevLen += len;
}
}
delete [] finalBuffer;
std::cout << "all is set";

return true;
}

int main(int argc, char *argv[])
{
if( argc < 3 )
{
std::cerr << argv[0] << " input.dcm output.dcm Resolution" << std::endl;
return 1;
}

const char *filename = argv[1];
const char *outfilename = argv[2];
char *res = argv[3];

int resolution = atoi(res);

gdcm::StreamImageWriter theStreamWriter;

std::ofstream of;
of.open( 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 = 0xffff;
uint16_t secondTag1 = 0xe0dd;
uint32_t thirdTag1 = 0x00000000;
//uint16_t fourthTag1 = 0xffff;
const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
char* tmpBuffer2 = new char[theBufferSize1];

```

```

memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
//memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
assert( of && !of.eof() && of.good() );
of.write(tmpBuffer2, theBufferSize);
of.flush();
assert( of );

return 0;
}

```

27.131 TestByteSwap.cxx

This is a C++ example on how to use `gdcm::ByteSwap`

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.h"

#include <string.h> // memcpy

int myfunc()
{
    char vl_str[4];
    const char raw[] = "\000\000\000\004";
    memcpy(vl_str, raw, 4);
    uint32_t vl;
    gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem
        ((uint32_t*)&vl_str, gdcm::SwapCode::BigEndian, 1);
    memcpy(&vl, vl_str, 4);
    if( vl != 0x00000004 )
    {
        std::cerr << std::hex << "vl: " << vl << std::endl;
        return 1;
    }

    gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem
        (vl, gdcm::SwapCode::LittleEndian);
    if( vl != 0x00000004 )
    {
        std::cerr << std::hex << "vl: " << vl << std::endl;
        return 1;
    }

    gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem
        (vl, gdcm::SwapCode::BigEndian);
    std::cout << std::hex << "vl: " << vl << std::endl;
    if( vl != 0x4000000 )
    {
        return 1;
    }

    return 0;
}

int TestByteSwap(int , char *[])
{
    gdcm::SwapCode sc = gdcm::SwapCode::Unknown
        ;
    if ( gdcm::ByteSwap<uint16_t>::SystemIsBigEndian
        () )
    {

```

```

    sc = gdc::SwapCode::BigEndian;
}
else if ( gdc::ByteSwap<uint16_t>::SystemIsLittleEndian
() )
{
    sc = gdc::SwapCode::LittleEndian;
}
if( sc == gdc::SwapCode::Unknown )
{
    return 1;
}

std::cout << "sc: " << sc << std::endl;

uint16_t t = 0x1234;
gdc::ByteSwap<uint16_t>::SwapFromSwapCodeIntoSystem
(t, sc);
if( sc == gdc::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 == gdc::SwapCode::LittleEndian )
{
    if( t != 0x1234 )
    {
        std::cerr << std::hex << "t: " << t << std::endl;
        return 1;
    }
}

char n[2];
memcpy(n, &t, 2 );
gdc::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
((uint16_t*)n, sc, 1);
uint16_t tn = *((uint16_t*)n);
if( sc == gdc::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 == gdc::SwapCode::LittleEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
gdc::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
((uint16_t*)n, gdc::SwapCode::BigEndian, 1);
tn = *((uint16_t*)n);
if( sc == gdc::SwapCode::LittleEndian )
{
    if( tn != 0x3412 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
else if ( sc == gdc::SwapCode::BigEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}

if( myfunc() )
{

```

```

        return 1;
    }

    uint16_t array[] = { 0x1234 };
    gdcmm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
        (array,
         gdcmm::SwapCode::BigEndian,2);
    if ( array[0] != 0x3412 )
    {
        return 1;
    }

    return 0;
}

```

27.132 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( 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;
}

```

27.133 TestReader.py

This is a C++ example on how to use gdcmm::Reader

```

#####
#
#   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.
#
#####

import gdcmm
import os,sys

def TestRead(filename, verbose = False):
    r = gdcmm.Reader()
    r.SetFileName( filename )
    success = r.Read()
    #if verbose: print r.GetFile()
    if verbose: print r.GetFile().GetDataSet()
    return success

if __name__ == "__main__":
    success = 0
    try:
        filename = os.sys.argv[1]
        success += TestRead( filename, True )
    except:
        # loop over all files:
        gdcmm.Trace.DebugOff()
        gdcmm.Trace.WarningOff()
        t = gdcmm.Testing()
        nfiles = t.GetNumberOfFileNames()
        for i in range(0,nfiles):
            filename = t.GetFileName(i)
            success += TestRead( filename )

    # Test succeed ?
    sys.exit(success == 0)

```

27.134 threadgdcmm.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 "gdcmmImageReader.h"
#include "gdcmmDirectory.h"
#include "gdcmmSystem.h"

#include "vtkImageData.h"
#include "vtkStructuredPointsWriter.h"

#include <pthread.h>

struct threadparams
{
    const char **filenames;
    unsigned int nfiles;
    char *scalarpinter;
// 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 unsigned int 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;

        gdcmm::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 gdcmm::Image &image = reader.GetImage();
        unsigned long len = image.GetBufferLength();
        char * pointer = params->scalarpinter;
    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(unsigned int 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], nfiles);

    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();
    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 unsigned int 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
unsigned int 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();
writer->SetInput( output );
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 unsigned int 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 unsigned int nfiles = argc - 1;
    ReadFiles(nfiles, filenames);
}

return 0;
}

```

27.135 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 unsigned int 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;
}

```

27.136 uid_unique.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>
#include <set>

int main()
{
    gdcm::UIDGenerator uid;
    //const char myroot[] =
    "9876543210.9876543210.9876543210.9876543210.9876543210"; // fails in ~40000 tries
    const char myroot[] = "9876543210.9876543210.9876543210";
    uid.SetRoot( myroot );
    std::set<std::string> uids;
    uint64_t wrap = 0;
    uint64_t c = 0;
    while(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 );
    }
}

```

```

    }
    return 0;
}

```

27.137 VolumeSorter.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
*/
#include "gdcmsorter.h"
#include "gdcmIPPSorter.h"
#include "gdcmscanner.h"
#include "gdcmdataset.h"
#include "gdcmattribute.h"
#include "gdcmtesting.h"

bool mysort1(gdcms::DataSet const & ds1, gdcms::DataSet
             const & ds2 )
{
    gdcms::Attribute<0x0020,0x000d> at1;
    at1.Set( ds1 );
    gdcms::Attribute<0x0020,0x000d> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort2(gdcms::DataSet const & ds1, gdcms::DataSet
             const & ds2 )
{
    gdcms::Attribute<0x0020,0x000e> at1;
    at1.Set( ds1 );
    gdcms::Attribute<0x0020,0x000e> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort3(gdcms::DataSet const & ds1, gdcms::DataSet
             const & ds2 )
{
    // This is a floating point number is the comparison ok ?
    gdcms::Attribute<0x0020,0x0037> at1;
    at1.Set( ds1 );
    gdcms::Attribute<0x0020,0x0037> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort4(gdcms::DataSet const & ds1, gdcms::DataSet
             const & ds2 )
{
    // Do the IPP sorting here
    gdcms::Attribute<0x0020,0x0032> ipp1;
    gdcms::Attribute<0x0020,0x0037> iop1;
    ipp1.Set( ds1 );
    iop1.Set( ds1 );
    gdcms::Attribute<0x0020,0x0032> ipp2;
    gdcms::Attribute<0x0020,0x0037> iop2;
    ipp2.Set( ds2 );
    iop2.Set( ds2 );
    if( iop1 != iop2 )
    {
        return false;
    }
}

```

```

    // else
    double normal[3];
    normal[0] = iopl[1]*iopl[5] - iopl[2]*iopl[4];
    normal[1] = iopl[2]*iopl[3] - iopl[0]*iopl[5];
    normal[2] = iopl[0]*iopl[4] - iopl[1]*iopl[3];
    double dist1 = 0;
    for (int i = 0; i < 3; ++i) dist1 += normal[i]*ipp1[i];
    double dist2 = 0;
    for (int i = 0; i < 3; ++i) dist2 += normal[i]*ipp2[i];

    std::cout << dist1 << ", " << dist2 << std::endl;
    return dist1 < dist2;
}

int main(int argc, char *argv[])
{
    const char *extradataroot = gdcm::Testing::GetDataExtraRoot
        ();
    std::string dirl;
    if( argc < 2 )
    {
        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 !
    const gdcm::Directory::FileNamesType &l1 = d.
        GetFileNames();
    const unsigned int nfiles = l1.size();
    std::cout << nfiles << std::endl;

    //if( nfiles != 280 )
    // {
    //     return 1;
    // }

    //d.Print( std::cout );

    gdcm::Scanner s;
    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
    s.AddTag( t1 );
    s.AddTag( t2 );
    //s.AddTag( t3 );
    //s.AddTag( t4 );
    //s.AddTag( t5 );
    //s.AddTag( t6 );
    bool b = s.Scan( d.GetFileNames() );
    if( !b )
    {
        std::cerr << "Scanner failed" << std::endl;
        return 1;
    }

    //s.Print( std::cout );

    // Only get the DICOM files:
    gdcm::Directory::FileNamesType l2 = s.GetKeys
        ();
    const unsigned int nfiles2 = l2.size();
    std::cout << nfiles2 << std::endl;

    if ( nfiles2 > nfiles )
    {
        return 1;
    }

    gdcm::Sorter sorter;

```

```

    sorter.SetSortFunction( mysort1 );
    sorter.StableSort( 12 );

    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:
    unsigned int nvalues = 0;
{
    gdcm::Scanner s;
    s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( d.GetFileNames() );

    //s.Print( std::cout );

    const gdcm::Scanner::ValueType &values = s.
        GetValues();
    nvalues = values.size();
    std::cout << "There are " << nvalues << " different type of values" <<
        std::endl;
    assert( nfiles2 % nvalues == 0 );
    std::cout << "Series is composed of " << (nfiles/nvalues) << " different 3D
        volumes" << std::endl;
}

gdcm::Directory::FileNamesType sorted_files =
    sorter.GetFileNames();

    // Which means we can take nvalues files at a time and execute
    // gdcm::IPPSorter on it:
    gdcm::IPPSorter ippsorter;
    gdcm::Directory::FileNamesType sub(
        sorted_files.begin(), sorted_files.begin() + nvalues);
    std::cout << sub.size() << std::endl;
    std::cout << sub[0] << std::endl;
    std::cout << sub[nvalues-1] << std::endl;
    ippsorter.SetComputeZSpacing( false );
    if( !ippsorter.Sort( sub ) )
    {
        std::cerr << "Could not sort" << std::endl;
        return 1;
    }

    std::cout << "IPPSorter:" << std::endl;
    ippsorter.Print( std::cout );

    return 0;
}

```

27.138 WriteBuffer.py

```

#####
#
#   Program: GDCM (Grassroots DICOM). A DICOM library
#
#   Copyright (c) 2006-2011 Mathieu Malaterre
#   All rights reserved.
#   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#   This software is distributed 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:

http://chuckhahm.com/Ischem/Zurich/XX_0134

(2005,1132) SQ (Sequence with undefined length #=8)      # u/1, 1 Unknown Tag &
    Data
    (fffe,e000) na (Item with undefined length #=9)      # u/1, 1 Item
        (2005,0011) LO [Philips MR Imaging DD 002]      # 26, 1
            PrivateCreator
            (2005,1137) PN [PDF_CONTROL_GEN_PARS]        # 20, 1 Unknown
                Tag & Data
                (2005,1138) PN (no value available)      # 0, 0 Unknown
                    Tag & Data
                    (2005,1139) PN [IEEE_PDF]            # 8, 1 Unknown
                        Tag & Data
                        (2005,1140) PN (no value available) # 0, 0 Unknown
                            Tag & Data
                            (2005,1141) PN (no value available) # 0, 0 Unknown
                                Tag & Data
                                (2005,1143) SL 3103      # 4, 1 Unknown
                                    Tag & Data
                                    (2005,1144) OW
                                        0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008... # 3104, 1 Unknown Tag & Data
                                        (2005,1147) CS [Y] # 2, 1 Unknown
                                            Tag & Data
                                            (fffe,e00d) na (ItemDelimitationItem) # 0, 0
                                                ItemDelimitationItem
                                                (fffe,e000) na (Item with undefined length #=9) # u/1, 1 Item
                                                    (2005,0011) LO [Philips MR Imaging DD 002] # 26, 1
                                                        PrivateCreator
                                                        (2005,1137) PN [PDF_CONTROL_PREP_PARS] # 22, 1 Unknown
                                                            Tag & Data
                                                            (2005,1138) PN (no value available) # 0, 0 Unknown
                                                                Tag & Data
                                                                (2005,1139) PN [IEEE_PDF] # 8, 1 Unknown
                                                                    Tag & Data
                                                                    (2005,1140) PN (no value available) # 0, 0 Unknown
                                                                        Tag & Data
                                                                        (2005,1141) PN (no value available) # 0, 0 Unknown
                                                                            Tag & Data
                                                                            (2005,1143) SL 7934 # 4, 1 Unknown
                                                                                Tag & Data
                                                                                (2005,1144) OW
                                                                                    19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008... # 7934, 1 Unknown Tag & Data
                                                                                    (2005,1147) CS [Y] # 2, 1 Unknown
                                                                                        Tag & Data
                                                                                        (fffe,e00d) na (ItemDelimitationItem) # 0, 0
                                                                                            ItemDelimitationItem
...
"""

import sys
import gdcm

if __name__ == "__main__":

    file1 = sys.argv[1]
    file2 = sys.argv[2]

    r = gdcm.Reader()
    r.SetFileName( file1 )
    if not r.Read():
        sys.exit(1)

    fg = gdcm.FilenameGenerator()
    f = r.GetFile()
    ds = f.GetDataSet()
    tsis = gdcm.Tag(0x2005,0x1132) #
    if ds.FindDataElement( tsis ):
        sis = ds.GetDataElement( tsis )
        #sqsis = sis.GetSequenceOfItems()
        # GetValueAsSQ handle more cases
        sqsis = sis.GetValueAsSQ()
        if sqsis.GetNumberOfItems():
            nitems = sqsis.GetNumberOfItems();
            fg.SetNumberOfFileNames( nitems )
            fg.SetPrefix( file2 )
            if not fg.Generate():
                print "problem"
                sys.exit(1)

```

```
for i in range(0,nitems):
    item1 = sqsis.GetItem(i+1) # Item start at 1
    nestedds = item1.GetNestedDataSet()
    tprcs = gdcM.Tag(0x2005,0x1144) #
    if nestedds.FindDataElement( tprcs ):
        prcs = nestedds.GetDataElement( tprcs )
        bv = prcs.GetByteValue()
        print bv
        f = open( fg.GetFilename(i) , "w" )
        f.write( bv.WriteBuffer() )
```

Index

- ~ASN1
 - gdcmm::ASN1, 161
- ~AnonymizeEvent
 - gdcmm::AnonymizeEvent, 146
- ~Anonymizer
 - gdcmm::Anonymizer, 150
- ~Attribute
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 177
- ~AudioCodec
 - gdcmm::AudioCodec, 187
- ~Base64
 - gdcmm::Base64, 188
- ~BasePDU
 - gdcmm::network::BasePDU, 192
- ~BaseRootQuery
 - gdcmm::BaseRootQuery, 195
- ~Bitmap
 - gdcmm::Bitmap, 204
- ~BitmapToBitmapFilter
 - gdcmm::BitmapToBitmapFilter, 210
- ~ByteSwapFilter
 - gdcmm::ByteSwapFilter, 214
- ~ByteValue
 - gdcmm::ByteValue, 216
- ~CSAHeader
 - gdcmm::CSAHeader, 253
- ~Coder
 - gdcmm::Coder, 231
- ~Command
 - gdcmm::Command, 236
- ~CommandDataSet
 - gdcmm::CommandDataSet, 238
- ~CryptographicMessageSyntax
 - gdcmm::CryptographicMessageSyntax, 246
- ~Curve
 - gdcmm::Curve, 264
- ~DICOMDIRGenerator
 - gdcmm::DICOMDIRGenerator, 295
- ~DataEvent
 - gdcmm::DataEvent, 277
- ~DataSetEvent
 - gdcmm::DataSetEvent, 286
- ~Decoder
 - gdcmm::Decoder, 288
- ~Defs
 - gdcmm::Defs, 290
- ~DeltaEncodingCodec
 - gdcmm::DeltaEncodingCodec, 293
- ~DictConverter
 - gdcmm::DictConverter, 300
- ~DictPrinter
 - gdcmm::DictPrinter, 305
- ~Dicts
 - gdcmm::Dicts, 307
- ~DirectionCosines
 - gdcmm::DirectionCosines, 310
- ~Directory
 - gdcmm::Directory, 313
- ~Dumper
 - gdcmm::Dumper, 317
- ~Element
 - gdcmm::Element< TVR, VM::VM1_n >, 324
- ~Event
 - gdcmm::Event, 341
- ~Exception
 - gdcmm::Exception, 343
- ~File
 - gdcmm::File, 351
- ~FileDerivation
 - gdcmm::FileDerivation, 353
- ~FileExplicitFilter
 - gdcmm::FileExplicitFilter, 356
- ~FileMetaInformation
 - gdcmm::FileMetaInformation, 359
- ~FilenameGenerator
 - gdcmm::FilenameGenerator, 365
- ~GDCMImageIO2
 - itk::GDCMImageIO2, 380
- ~Global
 - gdcmm::Global, 385
- ~GroupDict
 - gdcmm::GroupDict, 387
- ~IPPSorter
 - gdcmm::IPPSorter, 441
- ~IconImageFilter
 - gdcmm::IconImageFilter, 389
- ~IconImageGenerator
 - gdcmm::IconImageGenerator, 391
- ~Image

- gdcmm::Image, 395
- ~ImageApplyLookupTable
 - gdcmm::ImageApplyLookupTable, 399
- ~ImageChangePhotometricInterpretation
 - gdcmm::ImageChangePhotometricInterpretation, 402
- ~ImageChangePlanarConfiguration
 - gdcmm::ImageChangePlanarConfiguration, 405
- ~ImageChangeTransferSyntax
 - gdcmm::ImageChangeTransferSyntax, 408
- ~ImageCodec
 - gdcmm::ImageCodec, 412
- ~ImageConverter
 - gdcmm::ImageConverter, 416
- ~ImageFragmentSplitter
 - gdcmm::ImageFragmentSplitter, 418
- ~ImageReader
 - gdcmm::ImageReader, 424
- ~ImageToImageFilter
 - gdcmm::ImageToImageFilter, 427
- ~ImageWriter
 - gdcmm::ImageWriter, 429
- ~JPEG12Codec
 - gdcmm::JPEG12Codec, 449
- ~JPEG16Codec
 - gdcmm::JPEG16Codec, 451
- ~JPEG2000Codec
 - gdcmm::JPEG2000Codec, 453
- ~JPEG8Codec
 - gdcmm::JPEG8Codec, 456
- ~JPEGCodec
 - gdcmm::JPEGCodec, 458
- ~JPEGLSCodec
 - gdcmm::JPEGLSCodec, 462
- ~KAKADUCodec
 - gdcmm::KAKADUCodec, 465
- ~LookupTable
 - gdcmm::LookupTable, 469
- ~MD5
 - gdcmm::MD5, 476
- ~MemberCommand
 - gdcmm::MemberCommand, 485
- ~MeshPrimitive
 - gdcmm::MeshPrimitive, 489
- ~ModuleEntry
 - gdcmm::ModuleEntry, 495
- ~Object
 - gdcmm::Object, 507
- ~Orientation
 - gdcmm::Orientation, 509
- ~Overlay
 - gdcmm::Overlay, 513
- ~PDBHeader
 - gdcmm::PDBHeader, 525
- ~PDFCodec
 - gdcmm::PDFCodec, 527
- ~PNMCodec
 - gdcmm::PNMCodec, 551
- ~PVRGCodec
 - gdcmm::PVRGCodec, 572
- ~ParseException
 - gdcmm::ParseException, 517
- ~Parser
 - gdcmm::Parser, 519
- ~PixelFormat
 - gdcmm::PixelFormat, 535
- ~Pixmap
 - gdcmm::Pixmap, 540
- ~PixmapReader
 - gdcmm::PixmapReader, 543
- ~PixmapToPixmapFilter
 - gdcmm::PixmapToPixmapFilter, 545
- ~PixmapWriter
 - gdcmm::PixmapWriter, 548
- ~Preamble
 - gdcmm::Preamble, 552
- ~Printer
 - gdcmm::Printer, 564
- ~PrivateDict
 - gdcmm::PrivateDict, 566
- ~ProgressEvent
 - gdcmm::ProgressEvent, 570
- ~PythonFilter
 - gdcmm::PythonFilter, 573
- ~QueryBase
 - gdcmm::QueryBase, 575
- ~RAWCodec
 - gdcmm::RAWCodec, 585
- ~RLECodec
 - gdcmm::RLECodec, 595
- ~Reader
 - gdcmm::Reader, 589
- ~Rescaler
 - gdcmm::Rescaler, 592
- ~SHA1
 - gdcmm::SHA1, 633
- ~Scanner
 - gdcmm::Scanner, 601
- ~Segment
 - gdcmm::Segment, 606
- ~SegmentReader
 - gdcmm::SegmentReader, 611
- ~SegmentWriter
 - gdcmm::SegmentWriter, 614
- ~SegmentedPaletteColorLookupTable
 - gdcmm::SegmentedPaletteColorLookupTable, 609
- ~SerieHelper
 - gdcmm::SerieHelper, 626
- ~ServiceClassUser

- gdcmm::ServiceClassUser, 630
- ~SimpleMemberCommand
 - gdcmm::SimpleMemberCommand, 636
- ~SimpleSubjectWatcher
 - gdcmm::SimpleSubjectWatcher, 638
- ~SmartPointer
 - gdcmm::SmartPointer, 641
- ~Sorter
 - gdcmm::Sorter, 645
- ~Spacing
 - gdcmm::Spacing, 648
- ~SplitMosaicFilter
 - gdcmm::SplitMosaicFilter, 650
- ~StreamImageReader
 - gdcmm::StreamImageReader, 654
- ~StreamImageWriter
 - gdcmm::StreamImageWriter, 659
- ~StringFilter
 - gdcmm::StringFilter, 666
- ~Subject
 - gdcmm::Subject, 669
- ~Surface
 - gdcmm::Surface, 674
- ~SurfaceReader
 - gdcmm::SurfaceReader, 681
- ~SurfaceWriter
 - gdcmm::SurfaceWriter, 683
- ~Table
 - gdcmm::Table, 691
- ~TableEntry
 - gdcmm::TableEntry, 692
- ~TableReader
 - gdcmm::TableReader, 693
- ~TagPath
 - gdcmm::TagPath, 702
- ~Testing
 - gdcmm::Testing, 704
- ~Trace
 - gdcmm::Trace, 707
- ~Transition
 - gdcmm::network::Transition, 714
- ~ULAction
 - gdcmm::network::ULAction, 740
- ~ULBasicCallback
 - gdcmm::network::ULBasicCallback, 774
- ~ULConnection
 - gdcmm::network::ULConnection, 776
- ~ULConnectionCallback
 - gdcmm::network::ULConnectionCallback, 778
- ~ULConnectionManager
 - gdcmm::network::ULConnectionManager, 781
- ~ULEvent
 - gdcmm::network::ULEvent, 783
- ~ULWritingCallback
 - gdcmm::network::ULWritingCallback, 785
- ~Validate
 - gdcmm::Validate, 795
- ~Value
 - gdcmm::Value, 797
- ~Version
 - gdcmm::Version, 798
- ~Writer
 - gdcmm::Writer, 858
- ~XMLDictReader
 - gdcmm::XMLDictReader, 861
- ~XMLPrivateDictReader
 - gdcmm::XMLPrivateDictReader, 863
- ~vtkGDCMImageReader
 - vtkGDCMImageReader, 815
- ~vtkGDCMImageWriter
 - vtkGDCMImageWriter, 820
- ~vtkGDCMMedicalImageProperties
 - vtkGDCMMedicalImageProperties, 823
- ~vtkGDCMPolyDataReader
 - vtkGDCMPolyDataReader, 825
- ~vtkGDCMPolyDataWriter
 - vtkGDCMPolyDataWriter, 828
- ~vtkGDCMTesting
 - vtkGDCMTesting, 830
- ~vtkGDCMThreadedImageReader
 - vtkGDCMThreadedImageReader, 832
- ~vtkGDCMThreadedImageReader2
 - vtkGDCMThreadedImageReader2, 834
- ~vtkImageColorViewer
 - vtkImageColorViewer, 838
- ~vtkImageMapToColors16
 - vtkImageMapToColors16, 843
- ~vtkImageMapToWindowLevelColors2
 - vtkImageMapToWindowLevelColors2, 845
- ~vtkImagePlanarComponentsToComponents
 - vtkImagePlanarComponentsToComponents, 846
- ~vtkImageRGBToYBR
 - vtkImageRGBToYBR, 847
- ~vtkImageYBRToRGB
 - vtkImageYBRToRGB, 848
- ~vtkLookupTable16
 - vtkLookupTable16, 849
- ~vtkRTStructSetProperties
 - vtkRTStructSetProperties, 851
- AE
 - gdcmm::VR, 807
- AES128_CIPHER
 - gdcmm::CryptographicMessageSyntax, 246
- AES192_CIPHER
 - gdcmm::CryptographicMessageSyntax, 246
- AES256_CIPHER
 - gdcmm::CryptographicMessageSyntax, 246

- ALGOType_END
 - gdcmm::Segment, 606
- ARGB
 - gdcmm::PhotometricInterpretation, 532
- AS
 - gdcmm::VR, 807
- AT
 - gdcmm::VR, 807
- AUTOMATIC
 - gdcmm::Segment, 606
- AXIAL
 - gdcmm::Orientation, 509
- AAabortPDU
 - gdcmm::network::AAabortPDU, 134
- AAAssociateACPDU
 - gdcmm::network::AAAssociateACPDU, 137
- AAAssociateRJPDU
 - gdcmm::network::AAAssociateRJPDU, 139
- AAAssociateRQPDU
 - gdcmm::network::AAAssociateACPDU, 137
 - gdcmm::network::AAAssociateRQPDU, 141
- AECOMP
 - gdcmm, 117
- ALGOType
 - gdcmm::Segment, 606
- ARTIMTimer
 - gdcmm::network::ARTIMTimer, 160
- AREleaseRPPDU
 - gdcmm::network::AREleaseRPPDU, 157
- AREleaseRQPDU
 - gdcmm::network::AREleaseRQPDU, 158
- ASCOMP
 - gdcmm, 117
- ASN1
 - gdcmm::ASN1, 161
- AbstractSyntax
 - gdcmm::network::AbstractSyntax, 144
- ActiveComponent
 - vtkImageMapToColors16, 844
- Add
 - gdcmm::GroupDict, 387
- AddAcceptedPresentationContext
 - gdcmm::network::ULConnection, 776
- AddCSAHeaderDictEntry
 - gdcmm::CSAHeaderDict, 256
- AddContourReferencedFrameOfReference
 - vtkRTStructSetProperties, 851
- AddDerivationDescription
 - gdcmm::FileDerivation, 353
- AddDictEntry
 - gdcmm::Dict, 297
 - gdcmm::PrivateDict, 566
- AddFile
 - gdcmm::FileSet, 367
- gdcmm::SerieHelper, 626
- AddFileName
 - gdcmm::SerieHelper, 626
- AddFragment
 - gdcmm::SequenceOfFragments, 617
- AddGroupLength
 - gdcmm::DictConverter, 300
- AddIOD
 - gdcmm::IODs, 439
- AddIODEntry
 - gdcmm::IOD, 436
- AddImageDirectoryRecord
 - gdcmm::DICOMDIRGenerator, 295
- AddInput
 - vtkImageColorViewer, 838
- AddInputConnection
 - vtkImageColorViewer, 838
- AddItem
 - gdcmm::SequenceOfItems, 622
- AddMacro
 - gdcmm::Macros, 474
 - gdcmm::Module, 492
- AddMacroEntry
 - gdcmm::Macro, 473
- AddModule
 - gdcmm::Modules, 497
- AddModuleEntry
 - gdcmm::Module, 492
 - gdcmm::NestedModuleEntries, 504
- AddObserver
 - gdcmm::Subject, 669, 670
- AddPatientDirectoryRecord
 - gdcmm::DICOMDIRGenerator, 295
- AddPresentationContext
 - gdcmm::network::AAAssociateRQPDU, 141
 - gdcmm::PresentationContextGenerator, 557
- AddPresentationContextAC
 - gdcmm::network::AAAssociateACPDU, 137
- AddPresentationDataValue
 - gdcmm::network::PDataTFPDU, 521
- AddPrimitiveData
 - gdcmm::MeshPrimitive, 489
- AddPrivateTag
 - gdcmm::Scanner, 601
- AddPurposeOfReferenceCodeSequence
 - gdcmm::FileDerivation, 353
- AddQueryDataSet
 - gdcmm::BaseRootQuery, 195
- AddReference
 - gdcmm::FileDerivation, 353
- AddReferencedFrameOfReference
 - vtkRTStructSetProperties, 851
- AddRestriction
 - gdcmm::SerieHelper, 626

- AddSegment
 - gdcm::SegmentWriter, 614
- AddSelect
 - gdcm::Sorter, 645
- AddSeriesDirectoryRecord
 - gdcm::DICOmdirGenerator, 295
- AddSkipTag
 - gdcm::Scanner, 601
- AddSourceImageSequence
 - gdcm::FileDerivation, 353
- AddStructureSetROI
 - vtkRTStructSetProperties, 851
- AddStructureSetROIObservation
 - vtkRTStructSetProperties, 852
- AddStudyDirectoryRecord
 - gdcm::DICOmdirGenerator, 295
- AddSurface
 - gdcm::Segment, 606
- AddTag
 - gdcm::Scanner, 601
- AddTransferSyntax
 - gdcm::network::PresentationContextRQ, 559
 - gdcm::PresentationContext, 554
- AffectedSOPClassUID
 - gdcm::network::CEchoRQ, 220
- Allocate
 - gdcm::LookupTable, 469
- AmbulatoryECGWaveformStorage
 - gdcm::MediaStorage, 480
 - gdcm::UIDs, 727
- AnatomicRegion
 - gdcm::Segment, 607
- AnonymizeEvent
 - gdcm::AnonymizeEvent, 146
- Anonymizer
 - gdcm::Anonymizer, 150
- Append
 - gdcm::Global, 385
- AppendImplementationClassUID
 - gdcm::FileMetaInformation, 359
- ApplicationContext
 - gdcm::network::ApplicationContext, 153
- Apply
 - gdcm::ImageApplyLookupTable, 399
- ApplyInverseVideo
 - vtkGDCMImageReader, 818
- ApplyLookupTable
 - vtkGDCMImageReader, 818
- ApplyPlanarConfiguration
 - vtkGDCMImageReader, 818
- ApplyShiftScale
 - vtkGDCMImageReader, 818
- ApplyYBRToRGB
 - vtkGDCMImageReader, 818
- AreOverlaysInPixelData
 - gdcm::Bitmap, 204
 - gdcm::Pixmap, 540
- ArrayIncludeMacrosType
 - gdcm::Macro, 472
 - gdcm::Module, 492
- ArrayType
 - gdcm::Attribute, 164
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 170
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 177
- AsynchronousOperationsWindowSub
 - gdcm::network::AsynchronousOperationsWindowSub, 162
- Attribute
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 177
 - gdcm::terminal, 130
- Audio
 - gdcm::MediaStorage, 481
- AudioSRStorageTrialRetired
 - gdcm::UIDs, 728
- AudioCodec
 - gdcm::AudioCodec, 187
- AutoPixelMinMax
 - gdcm::IconImageGenerator, 391
- BLUE
 - gdcm::LookupTable, 469
- BALCPPProtect
 - gdcm::Anonymizer, 150
- backslash
 - gdcm, 119
- BadBigEndian
 - gdcm::SwapCode, 685
- BadLittleEndian
 - gdcm::SwapCode, 684
- Base64
 - gdcm::Base64, 188
- BaseRootQuery
 - gdcm::BaseRootQuery, 195
- BasicAnnotationBoxSOPClass
 - gdcm::UIDs, 726
- BasicColorImageBoxSOPClass
 - gdcm::UIDs, 726
- BasicColorPrintManagementMetaSOPClass
 - gdcm::UIDs, 726
- BasicFilmBoxSOPClass
 - gdcm::UIDs, 726
- BasicFilmSessionSOPClass
 - gdcm::UIDs, 726
- BasicGrayscaleImageBoxSOPClass
 - gdcm::UIDs, 726

- BasicGrayscalePrintManagementMetaSOPClass
 - gdcm::UIDs, 726
- BasicPrintImageOverlayBoxSOPClassRetired
 - gdcm::UIDs, 727
- BasicStudyContentNotificationSOPClassRetired
 - gdcm::UIDs, 726
- BasicTextSR
 - gdcm::MediaStorage, 480
- BasicTextSRStorage
 - gdcm::UIDs, 728
- BasicVoiceAudioWaveformStorage
 - gdcm::MediaStorage, 480
 - gdcm::UIDs, 727
- BasicApplicationLevelConfidentialityProfile
 - gdcm::Anonymizer, 150
- BasicCodedEntry
 - gdcm::SegmentHelper::BasicCodedEntry, 198
- BasicOffsetTable
 - gdcm::BasicOffsetTable, 200
- Begin
 - gdcm::CSAHeaderDict, 257
 - gdcm::DataSet, 280
 - gdcm::Dict, 297
 - gdcm::IODs, 439
 - gdcm::Scanner, 601
 - gdcm::SequenceOfFragments, 617
 - gdcm::SequenceOfItems, 622
- BigEndian
 - gdcm::SwapCode, 684
- BitSample
 - gdcm::JPEGCodec, 460
 - gdcm::LookupTable, 471
- Bitmap
 - gdcm::Bitmap, 204
 - gdcm::JPEG2000Codec, 454
 - gdcm::PixelFormat, 537
- BitmapToBitmapFilter
 - gdcm::BitmapToBitmapFilter, 210
- black
 - gdcm::terminal, 130
- BlendingSoftcopyPresentationStateStorageSOPClass
 - gdcm::UIDs, 728
- blink
 - gdcm::terminal, 130
- blue
 - gdcm::terminal, 130
- BreakConnection
 - gdcm::network::ULConnectionManager, 781
- BreakConnectionNow
 - gdcm::network::ULConnectionManager, 781
- BreastImagingRelevantPatientInformationQuery
 - gdcm::UIDs, 729
- bright
 - gdcm::terminal, 130
- Build
 - vtkLookupTable16, 849
- ByteBuffer
 - gdcm::ByteBuffer, 212
- ByteSwap
 - gdcm::ByteSwapFilter, 214
- ByteSwapFilter
 - gdcm::ByteSwapFilter, 214
- ByteValue
 - gdcm::ByteValue, 216
- bytes
 - gdcm::Tag, 701
- C_CANCEL_RQ
 - gdcm::network::DIMSE, 309
- C_ECHO_RQ
 - gdcm::network::DIMSE, 309
- C_ECHO_RSP
 - gdcm::network::DIMSE, 309
- C_FIND_RQ
 - gdcm::network::DIMSE, 309
- C_FIND_RSP
 - gdcm::network::DIMSE, 309
- C_GET_RQ
 - gdcm::network::DIMSE, 309
- C_GET_RSP
 - gdcm::network::DIMSE, 309
- C_MOVE_RQ
 - gdcm::network::DIMSE, 309
- C_MOVE_RSP
 - gdcm::network::DIMSE, 309
- C_STORE_RQ
 - gdcm::network::DIMSE, 309
- C_STORE_RSP
 - gdcm::network::DIMSE, 309
- CALIBRATED
 - gdcm::Spacing, 648
- CMYK
 - gdcm::PhotometricInterpretation, 532
- CONDENSED_STYLE
 - gdcm::Printer, 564
- CONSOLE
 - gdcm::terminal, 130
- CORONAL
 - gdcm::Orientation, 509
- CS
 - gdcm::VR, 807
- CSANonImageStorage
 - gdcm::MediaStorage, 480
- CT_private_ELE
 - gdcm::TransferSyntax, 711
- CTImageStorage
 - gdcm::MediaStorage, 479
 - gdcm::UIDs, 727

- CEcho
 - gdcm::CompositeNetworkFunctions, 241
- CFind
 - gdcm::CompositeNetworkFunctions, 241
- CM
 - gdcm::SegmentHelper::BasicCodedEntry, 198
- cMaxEventID
 - gdcm::network, 129
- cMaxStateID
 - gdcm::network, 129
- CMove
 - gdcm::CompositeNetworkFunctions, 241
- CSAElement
 - gdcm::CSAElement, 249
- CSAHeader
 - gdcm::CSAHeader, 253
 - gdcm::DataSet, 284
- CSAHeaderDict
 - gdcm::CSAHeaderDict, 256
- CSAHeaderDictEntry
 - gdcm::CSAHeaderDictEntry, 258
- CSAHeaderType
 - gdcm::CSAHeader, 253
- CSComp
 - gdcm, 117
- CSD
 - gdcm::SegmentHelper::BasicCodedEntry, 198
- CSV
 - gdcm::SegmentHelper::BasicCodedEntry, 198
- CStore
 - gdcm::CompositeNetworkFunctions, 242
- CV
 - gdcm::SegmentHelper::BasicCodedEntry, 198
- CanCode
 - gdcm::AudioCodec, 187
 - gdcm::Coder, 231
 - gdcm::ImageCodec, 412
 - gdcm::JPEG2000Codec, 453
 - gdcm::JPEGCodec, 459
 - gdcm::JPEGLSCodec, 462
 - gdcm::KAKADUCodec, 465
 - gdcm::PDFCodec, 527
 - gdcm::PNMCodec, 551
 - gdcm::PVRGCodec, 572
 - gdcm::RAWCodec, 585
 - gdcm::RLECodec, 595
- CanDecode
 - gdcm::AudioCodec, 187
 - gdcm::Decoder, 288
 - gdcm::DeltaEncodingCodec, 293
 - gdcm::ImageCodec, 412
 - gdcm::JPEG2000Codec, 453
 - gdcm::JPEGCodec, 459
 - gdcm::JPEGLSCodec, 462
 - gdcm::KAKADUCodec, 465
 - gdcm::PDFCodec, 527
 - gdcm::PNMCodec, 551
 - gdcm::PVRGCodec, 572
 - gdcm::RAWCodec, 585
 - gdcm::RLECodec, 595
- CanDisplay
 - gdcm::VR, 808
- CanEmptyTag
 - gdcm::Anonymizer, 150
- CanRead
 - gdcm::Reader, 589
- CanReadFile
 - itk::GDCMImageIO2, 380
 - vtkGDCMImageReader, 815
- CanReadImage
 - gdcm::StreamImageReader, 654
- CanStoreLossy
 - gdcm::TransferSyntax, 711
- CanWriteFile
 - gdcm::StreamImageWriter, 659
 - itk::GDCMImageIO2, 380
- CardiacElectrophysiologyWaveformStorage
 - gdcm::MediaStorage, 480
 - gdcm::UIDs, 727
- CardiacRelevantPatientInformationQuery
 - gdcm::UIDs, 730
- Change
 - gdcm::FileExplicitFilter, 356
 - gdcm::ImageChangePhotometricInterpretation, 402
 - gdcm::ImageChangePlanarConfiguration, 405
 - gdcm::ImageChangeTransferSyntax, 408
- ChangeFMI
 - gdcm::FileExplicitFilter, 356
- ChangeMonochrome
 - gdcm::ImageChangePhotometricInterpretation, 402
- CharacterDataHandler
 - gdcm::TableReader, 693
 - gdcm::XMLDictReader, 861
 - gdcm::XMLPrivateDictReader, 863
- CheckEvent
 - gdcm::AnonymizeEvent, 146
 - gdcm::DataEvent, 277
 - gdcm::DataSetEvent, 286
 - gdcm::Event, 341
 - gdcm::ProgressEvent, 570
- CheckFileMetaInformationOff
 - gdcm::Writer, 858
- CheckFileMetaInformationOn
 - gdcm::Writer, 858
- ChestCADSRStorage
 - gdcm::UIDs, 729
- CipherTypes
 - gdcm::CryptographicMessageSyntax, 246

Clear

- gdcm::Bitmap, 204
- gdcm::ByteValue, 216
- gdcm::DataElement, 269
- gdcm::DataSet, 280
- gdcm::IOD, 436
- gdcm::IODs, 439
- gdcm::Item, 445
- gdcm::LookupTable, 470
- gdcm::Macro, 473
- gdcm::Macros, 474
- gdcm::Module, 492
- gdcm::Modules, 497
- gdcm::Preamble, 553
- gdcm::SequenceOfFragments, 617
- gdcm::SequenceOfItems, 622
- gdcm::SerieHelper, 627
- gdcm::Value, 797
- vtkGDCMMedicalImageProperties, 823
- vtkRTStructSetProperties, 852

ClearSkipTags

- gdcm::Scanner, 601

ClearTags

- gdcm::Scanner, 601

Code

- gdcm::Coder, 231
- gdcm::JPEG2000Codec, 453
- gdcm::JPEGCodec, 459
- gdcm::JPEGLSCodec, 462
- gdcm::KAKADUCodec, 465
- gdcm::PVRGCodec, 572
- gdcm::RAWCodec, 585
- gdcm::RLECodec, 596

CodeString

- gdcm::CodeString, 233

Color

- gdcm::terminal, 130

ColorSoftcopyPresentationStateStorageSOPClass

- gdcm::UIDs, 728

ColorArray

- gdcm::SurfaceHelper, 678

Command

- gdcm::Command, 236

CommandDataSet

- gdcm::CommandDataSet, 238

CommandTypes

- gdcm::network::DIMSE, 308

CompOperators

- gdcm, 117

Compatible

- gdcm::VM, 804
- gdcm::VR, 808

Component

- gdcm::PersonName, 530

ComprehensiveSR

- gdcm::MediaStorage, 480

ComprehensiveSRStorage

- gdcm::UIDs, 728

ComprehensiveSRStorageTrialRetired

- gdcm::UIDs, 728

CompressionTypes

- vtkGDCMImageWriter, 820

Compute

- gdcm::MD5, 476
- gdcm::SHA1, 633

ComputeByteLength

- gdcm::SequenceOfFragments, 617

ComputeDataElement

- gdcm::DataSet, 280

ComputeDataSetMediaStorageSOPClass

- gdcm::FileMetaInformation, 359

ComputeDataSetTransferSyntax

- gdcm::FileMetaInformation, 359

ComputeDistAlongNormal

- gdcm::DirectionCosines, 310

ComputeFile

- gdcm::MD5, 476
- gdcm::SHA1, 634

ComputeFileMD5

- gdcm::Testing, 704

ComputeGroupLength

- gdcm::DataSet, 281

ComputeInterceptSlopePixelType

- gdcm::Rescaler, 592

ComputeLength

- gdcm::SequenceOfFragments, 617
- gdcm::SequenceOfItems, 622

ComputeLossyFlag

- gdcm::Bitmap, 204

ComputeMD5

- gdcm::Testing, 704

ComputeMOSAICDimensions

- gdcm::SplitMosaicFilter, 650

ComputeNumberOfSurfaces

- gdcm::SurfaceWriter, 683

ComputeOffsetTable

- gdcm::JPEGCodec, 459

ComputePixelAspectRatioFromPixelSpacing

- gdcm::Spacing, 648

ComputePixelTypeFromMinMax

- gdcm::Rescaler, 593

ComputeSpacingFromImagePositionPatient

- gdcm::ImageHelper, 419

ComputeVR

- gdcm::DataSetHelper, 287

ComputeZSpacing

- gdcm::IPPSorter, 443

ComputedRadiographyImageStorage

- gdcm::MediaStorage, 479
- gdcm::UIDs, 727
- ConcatenatePDVBlobs
 - gdcm::network::PresentationDataValue, 561
- Conditional
 - gdcm::Usage, 791
- const
 - gdcm::SOPClassUIDToIOD, 642
- const_iterator
 - gdcm::CodeString, 233
 - gdcm::LO, 466
 - gdcm::String, 664
- const_reference
 - gdcm::CodeString, 233
 - gdcm::LO, 466
 - gdcm::String, 664
- const_reverse_iterator
 - gdcm::CodeString, 233
 - gdcm::LO, 466
 - gdcm::String, 664
- ConstCharWrapper
 - gdcm::ConstCharWrapper, 243
- ConstIterator
 - gdcm::CSAHeaderDict, 256
 - gdcm::DataSet, 280
 - gdcm::Dict, 297
 - gdcm::Scanner, 600
 - gdcm::SequenceOfFragments, 617
 - gdcm::SequenceOfItems, 622
- ConstructAbortPDU
 - gdcm::network::PDUFactory, 528
- ConstructCEchoRQ
 - gdcm::network::CompositeMessageFactory, 239
- ConstructCFindRQ
 - gdcm::network::CompositeMessageFactory, 239
- ConstructCMoveRQ
 - gdcm::network::CompositeMessageFactory, 239
- ConstructCStoreRQ
 - gdcm::network::CompositeMessageFactory, 239
- ConstructCStoreRSP
 - gdcm::network::CompositeMessageFactory, 240
- ConstructFromString
 - gdcm::TagPath, 702
- ConstructFromTagList
 - gdcm::TagPath, 702
- ConstructPDU
 - gdcm::network::PDUFactory, 528
- ConstructPDV
 - gdcm::network::BaseCompositeMessage, 191
 - gdcm::network::CEchoRQ, 220
 - gdcm::network::CEchoRSP, 221
 - gdcm::network::CFindCancelRQ, 222
 - gdcm::network::CFindRQ, 223
 - gdcm::network::CFindRSP, 225
 - gdcm::network::CMoveCancelRq, 226
 - gdcm::network::CMoveRQ, 227
 - gdcm::network::CMoveRSP, 229
 - gdcm::network::CStoreRQ, 261
 - gdcm::network::CStoreRSP, 262
- ConstructQuery
 - gdcm::CompositeNetworkFunctions, 242
- ConstructReleasePDU
 - gdcm::network::PDUFactory, 528
- ConstructorType
 - gdcm::Dicts, 306
- Convert
 - gdcm::DictConverter, 300
 - gdcm::ImageConverter, 416
- ConvertRGBToPaletteColor
 - gdcm::IconImageGenerator, 391
- ConvertToCXX
 - gdcm::DictConverter, 300
- ConvertToXML
 - gdcm::DictConverter, 300
- Create
 - gdcm::Preamble, 553
- CreateCEchoPDU
 - gdcm::network::PDUFactory, 529
- CreateCFindPDU
 - gdcm::network::PDUFactory, 529
- CreateCMovePDU
 - gdcm::network::PDUFactory, 529
- CreateCStoreRQPDU
 - gdcm::network::PDUFactory, 529
- CreateCStoreRSPPDU
 - gdcm::network::PDUFactory, 529
- CreateDefaultUniqueSeriesIdentifier
 - gdcm::SerieHelper, 627
- CreateUniqueSeriesIdentifier
 - gdcm::SerieHelper, 627
- Cross
 - gdcm::DirectionCosines, 310
- CrossDot
 - gdcm::DirectionCosines, 310
- CryptographicMessageSyntax
 - gdcm::CryptographicMessageSyntax, 246
- Curve
 - gdcm::Curve, 264
 - vtkGDCMImageReader, 818
- Curves
 - gdcm::Pixmap, 541
- cyan
 - gdcm::terminal, 130
- DA
 - gdcm::VR, 807
- DATASET_FORMAT
 - gdcm::CSAHeader, 253

- DES3_CIPHER
 - gdcm::CryptographicMessageSyntax, 246
- DES_CIPHER
 - gdcm::CryptographicMessageSyntax, 246
- DETECTOR
 - gdcm::Spacing, 648
- DICOMApplicationContextName
 - gdcm::UIDs, 726
- DICOMControlledTerminology
 - gdcm::UIDs, 726
- DICOMUIDRegistry
 - gdcm::UIDs, 726
- DICT_DEBUG
 - gdcm::DictConverter, 300
- DICT_DEFAULT
 - gdcm::DictConverter, 300
- DICT_XML
 - gdcm::DictConverter, 300
- DS
 - gdcm::VR, 807
- DT
 - gdcm::VR, 808
- DAComp
 - gdcm, 117
- DICOMDIR
 - gdcm::DICOMDIR, 294
- DICOMDIRGenerator
 - gdcm::DICOMDIRGenerator, 295
- DTComp
 - gdcm, 117
- DataElement
 - gdcm::DataElement, 268
- DataElementSet
 - gdcm::DataSet, 280
- DataElementType
 - gdcm::ModuleEntry, 496
- DataEvent
 - gdcm::DataEvent, 277
- DataField
 - gdcm::CSAElement, 251
- DataPtr
 - gdcm::CSAElement, 249
- DataSetEvent
 - gdcm::DataSetEvent, 286
- DataSetHandled
 - gdcm::network::ULConnectionCallback, 778
- DataSetHandles
 - gdcm::network::ULConnectionCallback, 778
- DataSetMS
 - gdcm::FileMetaInformation, 362
- DataSetTS
 - gdcm::FileMetaInformation, 362
- DataWasPassed
 - vtkImageMapToColors16, 844
- DebugOff
 - gdcm::Trace, 707
- DebugOn
 - gdcm::Trace, 707
- Decode
 - gdcm::AudioCodec, 188
 - gdcm::Base64, 188
 - gdcm::Curve, 264
 - gdcm::Decoder, 288
 - gdcm::DeltaEncodingCodec, 293
 - gdcm::ImageCodec, 412
 - gdcm::JPEG12Codec, 449
 - gdcm::JPEG16Codec, 451
 - gdcm::JPEG2000Codec, 454
 - gdcm::JPEG8Codec, 456
 - gdcm::JPEGCodec, 459
 - gdcm::JPEGLSCodec, 462, 463
 - gdcm::KAKADUCodec, 465
 - gdcm::LookupTable, 470
 - gdcm::Overlay, 513
 - gdcm::PDFCodec, 528
 - gdcm::PVRGCodec, 573
 - gdcm::RAWCodec, 585
 - gdcm::RLECodec, 596
- DecodeBytes
 - gdcm::RAWCodec, 585
- Decompress
 - gdcm::Overlay, 513
- Decrypt
 - gdcm::CryptographicMessageSyntax, 246
- DeepCopy
 - vtkRTStructSetProperties, 852
- Default
 - gdcm::FileMetaInformation, 359
- DefinePixelExtent
 - gdcm::StreamImageReader, 654
 - gdcm::StreamImageWriter, 660
- DefineProperBufferLength
 - gdcm::StreamImageReader, 655
 - gdcm::StreamImageWriter, 660
- DefinedTerms
 - gdcm::DefinedTerms, 289
- DeflatedExplicitVRLittleEndian
 - gdcm::TransferSyntax, 710
 - gdcm::UIDs, 724
- DeformableSpatialRegistrationStorage
 - gdcm::UIDs, 728
- Defs
 - gdcm::Defs, 290
- DeleteDirectory
 - gdcm::System, 687
- DeltaEncodingCodec
 - gdcm::DeltaEncodingCodec, 293
- Derive

- gdcM::FileDerivation, 354
- Description
 - gdcM::ModuleEntry, 495
- DescriptionField
 - gdcM::ModuleEntry, 496
- DetachedInterpretationManagementSOPClassRetired
 - gdcM::UIDs, 726
- DetachedPatientManagementMetaSOPClassRetired
 - gdcM::UIDs, 726
- DetachedPatientManagementSOPClass
 - gdcM::MediaStorage, 480
- DetachedPatientManagementSOPClassRetired
 - gdcM::UIDs, 726
- DetachedResultsManagementMetaSOPClassRetired
 - gdcM::UIDs, 726
- DetachedResultsManagementSOPClassRetired
 - gdcM::UIDs, 726
- DetachedStudyManagementMetaSOPClassRetired
 - gdcM::UIDs, 726
- DetachedStudyManagementSOPClass
 - gdcM::MediaStorage, 480
- DetachedStudyManagementSOPClassRetired
 - gdcM::UIDs, 726
- DetachedVisitManagementSOPClass
 - gdcM::MediaStorage, 480
- DetachedVisitManagementSOPClassRetired
 - gdcM::UIDs, 726
- DetailSRStorageTrialRetired
 - gdcM::UIDs, 728
- DetermineEventByPDU
 - gdcM::network::PDUFactory, 529
- dicomAETitle
 - gdcM::UIDs, 730
- dicomApplicationCluster
 - gdcM::UIDs, 730
- dicomAssociationAcceptor
 - gdcM::UIDs, 730
- dicomAssociationInitiator
 - gdcM::UIDs, 730
- dicomAuthorizedNodeCertificateReference
 - gdcM::UIDs, 730
- dicomConfigurationRoot
 - gdcM::UIDs, 730
- dicomDescription
 - gdcM::UIDs, 730
- dicomDevice
 - gdcM::UIDs, 730
- dicomDeviceName
 - gdcM::UIDs, 730
- dicomDeviceSerialNumber
 - gdcM::UIDs, 730
- dicomDevicesRoot
 - gdcM::UIDs, 730
- dicomHostname
 - gdcM::UIDs, 730
- dicomInstalled
 - gdcM::UIDs, 730
- dicomInstitutionAddress
 - gdcM::UIDs, 730
- dicomInstitutionDepartmentName
 - gdcM::UIDs, 730
- dicomInstitutionName
 - gdcM::UIDs, 730
- dicomIssuerOfPatientID
 - gdcM::UIDs, 730
- dicomManufacturer
 - gdcM::UIDs, 730
- dicomManufacturerModelName
 - gdcM::UIDs, 730
- dicomNetworkAE
 - gdcM::UIDs, 730
- dicomNetworkConnection
 - gdcM::UIDs, 731
- dicomNetworkConnectionReference
 - gdcM::UIDs, 730
- dicomPort
 - gdcM::UIDs, 730
- dicomPreferredCalledAETitle
 - gdcM::UIDs, 730
- dicomPreferredCallingAETitle
 - gdcM::UIDs, 730
- dicomPrimaryDeviceType
 - gdcM::UIDs, 730
- dicomRelatedDeviceReference
 - gdcM::UIDs, 730
- dicomSOPClass
 - gdcM::UIDs, 730
- dicomSoftwareVersion
 - gdcM::UIDs, 730
- dicomStationName
 - gdcM::UIDs, 730
- dicomSupportedCharacterSet
 - gdcM::UIDs, 730
- dicomTLSCyphersuite
 - gdcM::UIDs, 730
- dicomThisNodeCertificateReference
 - gdcM::UIDs, 730
- dicomTransferCapability
 - gdcM::UIDs, 731
- dicomTransferRole
 - gdcM::UIDs, 730
- dicomTransferSyntax
 - gdcM::UIDs, 730
- dicomUniqueAETitle
 - gdcM::UIDs, 731
- dicomUniqueAETitlesRegistryRoot
 - gdcM::UIDs, 730
- dicomVendorData

- gdcmm::UIDs, 730
- Dict
 - gdcmm::Dict, 297
- DictConverter
 - gdcmm::DictConverter, 300
- DictEntry
 - gdcmm::DictEntry, 302
- DictPrinter
 - gdcmm::DictPrinter, 305
- Dicts
 - gdcmm::CSAHeaderDict, 257
 - gdcmm::Dict, 298
 - gdcmm::Dicts, 307
 - gdcmm::PrivateDict, 566
- difference_type
 - gdcmm::CodeString, 233
 - gdcmm::LO, 466
 - gdcmm::String, 664
- DigitalIntraoralXRayImageStorageForPresentation
 - gdcmm::UIDs, 727
- DigitalIntraoralXRayImageStorageForProcessing
 - gdcmm::MediaStorage, 479
 - gdcmm::UIDs, 727
- DigitalIntraoralXrayImageStorageForPresentation
 - gdcmm::MediaStorage, 479
- DigitalMammographyImageStorageForPresentation
 - gdcmm::MediaStorage, 479
- DigitalMammographyImageStorageForProcessing
 - gdcmm::MediaStorage, 479
- DigitalMammographyXRayImageStorageForPresentation
 - gdcmm::UIDs, 727
- DigitalMammographyXRayImageStorageForProcessing
 - gdcmm::UIDs, 727
- DigitalXRayImageStorageForPresentation
 - gdcmm::MediaStorage, 479
 - gdcmm::UIDs, 727
- DigitalXRayImageStorageForProcessing
 - gdcmm::MediaStorage, 479
 - gdcmm::UIDs, 727
- dim
 - gdcmm::terminal, 130
- Dimensions
 - gdcmm::Bitmap, 208
 - gdcmm::ImageCodec, 414
- DirCosTolerance
 - gdcmm::IPPSorter, 443
- DirectionCosines
 - gdcmm::DirectionCosines, 310
 - vtkGDCMImageReader, 818
- Directory
 - gdcmm::Directory, 313
- DoByteSwap
 - gdcmm::ImageCodec, 413
- DolconImage
 - gdcmm::PixmapWriter, 548
- DoInvertMonochrome
 - gdcmm::ImageCodec, 413
- DoOverlayCleanup
 - gdcmm::ImageCodec, 413
- DoPaddedCompositePixelCode
 - gdcmm::ImageCodec, 413
- DoPlanarConfiguration
 - gdcmm::ImageCodec, 413
- DoSimpleCopy
 - gdcmm::ImageCodec, 413
- DoYBR
 - gdcmm::ImageCodec, 413
- Dot
 - gdcmm::DirectionCosines, 310
- Dumper
 - gdcmm::Dumper, 317
- DuplicateAttributeError
 - gdcmm::Parser, 518
- eAABORTPDUREceivedOpen
 - gdcmm::network, 128
- eAABORTRequest
 - gdcmm::network, 128
- eAASSOCIATE_RQPDUREceived
 - gdcmm::network, 128
- eAASSOCIATERequestLocalUser
 - gdcmm::network, 128
- eAASSOCIATEResponseAccept
 - gdcmm::network, 128
- eAASSOCIATEResponseReject
 - gdcmm::network, 128
- eARELEASE_RPPDUREceived
 - gdcmm::network, 128
- eARELEASE_RQPDUREceivedOpen
 - gdcmm::network, 128
- eARELEASERequest
 - gdcmm::network, 128
- eARELEASEResponse
 - gdcmm::network, 128
- eARTIMTimerExpired
 - gdcmm::network, 128
- eASSOCIATE_ACPDUREceived
 - gdcmm::network, 128
- eASSOCIATE_RJPDUREceived
 - gdcmm::network, 128
- eArabic
 - gdcmm, 118
- eCyrillic
 - gdcmm, 118
- EDGE
 - gdcmm::MeshPrimitive, 489
- eEventDoesNotExist
 - gdcmm::network, 128

- eFind
 - gdcM, 119
- eGB18030
 - gdcM, 118
- eGreek
 - gdcM, 118
- eHebrew
 - gdcM, 118
- eImageOrFrame
 - gdcM, 118
- eJapanese
 - gdcM, 118
- eJapaneseKanjiMultibyte
 - gdcM, 118
- eJapaneseSupplementaryKanjiMultibyte
 - gdcM, 118
- eKoreanHangulHanjaMultibyte
 - gdcM, 118
- eLatin1
 - gdcM, 118
- eLatin2
 - gdcM, 118
- eLatin3
 - gdcM, 118
- eLatin4
 - gdcM, 118
- eLatin5
 - gdcM, 118
- eMove
 - gdcM, 119
- ePDATATFPDU
 - gdcM::network, 128
- ePDATArequest
 - gdcM::network, 128
- ePatient
 - gdcM, 118
- ePatientRootType
 - gdcM, 119
- eSeries
 - gdcM, 118
- eSta10ReleaseCollisionAc
 - gdcM::network, 128
- eSta11ReleaseCollisionRq
 - gdcM::network, 128
- eSta12ReleaseCollisionAcLocal
 - gdcM::network, 128
- eSta13AwaitingClose
 - gdcM::network, 128
- eSta1Idle
 - gdcM::network, 128
- eSta2Open
 - gdcM::network, 128
- eSta3WaitLocalAssoc
 - gdcM::network, 128
- eSta4LocalAssocDone
 - gdcM::network, 128
- eSta5WaitRemoteAssoc
 - gdcM::network, 128
- eSta6TransferReady
 - gdcM::network, 128
- eSta7WaitRelease
 - gdcM::network, 128
- eSta8WaitLocalRelease
 - gdcM::network, 128
- eSta9ReleaseCollisionRqLocal
 - gdcM::network, 128
- eStaDoesNotExist
 - gdcM::network, 128
- eStudy
 - gdcM, 118
- eStudyRootType
 - gdcM, 119
- eThai
 - gdcM, 118
- eTransportConnConfirmLocal
 - gdcM::network, 128
- eTransportConnIndicLocal
 - gdcM::network, 128
- eTransportConnectionClosed
 - gdcM::network, 128
- eUTF8
 - gdcM, 118
- eUnrecognizedPDUReceived
 - gdcM::network, 128
- ECharSet
 - gdcM, 118
- EEventID
 - gdcM::network, 127
- EQueryLevel
 - gdcM, 118
- EQueryType
 - gdcM, 118
- ERootType
 - gdcM, 119
- EStateID
 - gdcM::network, 128
- elem
 - gdcM::SerieHelper::Rule, 597
- Element
 - gdcM::Element< TVR, VM::VM1_n >, 324
- Empty
 - gdcM::Anonymizer, 150
 - gdcM::DataElement, 269
- EncapsulatedCDASStorage
 - gdcM::UIDs, 729
- EncapsulatedPDFStorage
 - gdcM::MediaStorage, 480
 - gdcM::UIDs, 729

- EncapsulatedDocument
 - gdcm::EncapsulatedDocument, 336
- Encode
 - gdcm::Base64, 189
- EncodeBytes
 - gdcm::System, 687
- Encrypt
 - gdcm::CryptographicMessageSyntax, 246
- End
 - gdcm::CSAHeaderDict, 257
 - gdcm::DataSet, 281
 - gdcm::Dict, 297
 - gdcm::IODs, 439
 - gdcm::Scanner, 601
 - gdcm::SequenceOfFragments, 617
 - gdcm::SequenceOfItems, 622
- EndElement
 - gdcm::TableReader, 693
 - gdcm::XMLDictReader, 861
 - gdcm::XMLPrivateDictReader, 863
- EndElementHandler
 - gdcm::Parser, 518
- EndFilter
 - gdcm::SimpleSubjectWatcher, 638
- EnhancedCTImageStorage
 - gdcm::MediaStorage, 479
 - gdcm::UIDs, 727
- EnhancedMRImageStorage
 - gdcm::MediaStorage, 479
 - gdcm::UIDs, 727
- EnhancedSR
 - gdcm::MediaStorage, 480
- EnhancedSRStorage
 - gdcm::UIDs, 728
- EnhancedXAImageStorage
 - gdcm::MediaStorage, 481
 - gdcm::UIDs, 728
- EnhancedXRImageStorage
 - gdcm::UIDs, 728
- EnumeratedValues
 - gdcm::EnumeratedValues, 340
- ErrorOff
 - gdcm::Trace, 707
- ErrorOn
 - gdcm::Trace, 708
- ErrorType
 - gdcm::Parser, 518
- EstablishConnection
 - gdcm::network::ULConnectionManager, 781
- EstablishConnectionMove
 - gdcm::network::ULConnectionManager, 781
- Event
 - gdcm::Event, 341
- Exception
 - gdcm::Exception, 343
- Execute
 - gdcm::Command, 236
 - gdcm::MemberCommand, 485
 - gdcm::SimpleMemberCommand, 636
- ExecuteData
 - vtkGDCMImageReader, 815
 - vtkGDCMThreadedImageReader, 832
- ExecuteInformation
 - vtkGDCMImageReader, 815
 - vtkGDCMThreadedImageReader, 832
- ExecuteQuery
 - gdcm::StringFilter, 666
- Explicit
 - gdcm::TransferSyntax, 710
- ExplicitVRBigEndian
 - gdcm::TransferSyntax, 710
 - gdcm::UIDs, 724
- ExplicitVRLittleEndian
 - gdcm::TransferSyntax, 710
 - gdcm::UIDs, 724
- Explore
 - gdcm::Directory, 313
- Extract
 - gdcm::IconImageFilter, 389
- ExtractIconImages
 - gdcm::IconImageFilter, 389
- ExtractVeprolconImages
 - gdcm::IconImageFilter, 389
- F
 - gdcm::Printer, 565
 - gdcm::Reader, 591
 - gdcm::Validate, 795
- FACET
 - gdcm::MeshPrimitive, 489
- FD
 - gdcm::VR, 808
- FL
 - gdcm::VR, 808
- FLOAT16
 - gdcm::PixelFormat, 535
- FLOAT32
 - gdcm::PixelFormat, 535
- FLOAT64
 - gdcm::PixelFormat, 535
- Fiducials
 - gdcm::Fiducials, 349
- File
 - gdcm::File, 351
- FileDerivation
 - gdcm::FileDerivation, 353
- FileExists
 - gdcm::System, 687

- FileExplicitFilter
 - gdcm::FileExplicitFilter, 356
- FilesDirectory
 - gdcm::System, 688
- FilesSymlink
 - gdcm::System, 688
- FileList
 - gdcm, 117
- FileMetaInformation
 - gdcm::FileMetaInformation, 359
- FileName
 - vtkGDCMPolyDataReader, 826
- FileNameOrdering
 - gdcm::SerieHelper, 627
- FileNames
 - vtkGDCMImageReader, 818
- FileSet
 - gdcm::FileSet, 367
- FileSize
 - gdcm::System, 688
- FileTime
 - gdcm::System, 688
- FileType
 - gdcm::FileSet, 367
- FileWithName
 - gdcm::FileWithName, 369
- Filename
 - gdcm::Filename, 363
- filename
 - gdcm::FileWithName, 369
- FilenameGenerator
 - gdcm::FilenameGenerator, 365
- FilenameType
 - gdcm::DICOMDIRGenerator, 295
 - gdcm::Directory, 312
 - gdcm::FilenameGenerator, 365
- Filenames
 - gdcm::Sorter, 646
- FilenamesType
 - gdcm::DICOMDIRGenerator, 295
 - gdcm::Directory, 312
 - gdcm::FilenameGenerator, 365
- FilesType
 - gdcm::FileSet, 367
- Fill
 - gdcm::ByteValue, 216
- FillFromDataSet
 - gdcm::FileMetaInformation, 359
- FillMedicalImageInformation
 - vtkGDCMImageReader, 815
 - vtkGDCMPolyDataReader, 825
- FindCSAElementByName
 - gdcm::CSAHeader, 253
- FindContext
 - gdcm::network::ULConnection, 776
- FindDataElement
 - gdcm::DataSet, 281
 - gdcm::Item, 445
 - gdcm::SequenceOfItems, 622
- FindDictEntry
 - gdcm::PrivateDict, 566
- FindMacroEntry
 - gdcm::Macro, 473
- FindModuleEntryInMacros
 - gdcm::Module, 492
- FindNextDataElement
 - gdcm::DataSet, 281
- FindPDBelementByName
 - gdcm::PDBHeader, 525
- FindPatientRootQuery
 - gdcm::FindPatientRootQuery, 371
- FindStudyRootQuery
 - gdcm::FindStudyRootQuery, 373
- FirstRender
 - vtkImageColorViewer, 841
- ForceRescale
 - vtkGDCMImageReader, 818
- FormatDateTime
 - gdcm::System, 688
- Fragment
 - gdcm::Fragment, 376
- FragmentVector
 - gdcm::SequenceOfFragments, 617
- FromString
 - gdcm::StringFilter, 667
- GDCM_DIFFERENT
 - gdcm, 117
- GDCM_EQUAL
 - gdcm, 117
- GDCM_GREATER
 - gdcm, 118
- GDCM_GREATEROREQUAL
 - gdcm, 118
- GDCM_LESS
 - gdcm, 118
- GDCM_LESSCOREQUAL
 - gdcm, 118
- GEMS
 - gdcm::Dicts, 306
- GEPrivate3DModelStorage
 - gdcm::MediaStorage, 480
- GRAY
 - gdcm::LookupTable, 469
- GREEN
 - gdcm::LookupTable, 469
- GDCM_DO_JOIN
 - gdcmStaticAssert.h, 925

- GDCM_DO_JOIN2
 - gdcmStaticAssert.h, 925
- GDCM_EXPORT
 - gdcmWin32.h, 946
- GDCM_FUNCTION
 - gdcmTrace.h, 932
- GDCM_JOIN
 - gdcmStaticAssert.h, 925
- GDCM_LEGACY
 - gdcmLegacyMacro.h, 901
- GDCM_LEGACY_BODY
 - gdcmLegacyMacro.h, 901
- GDCM_STATIC_ASSERT
 - gdcm::Attribute, 164
 - gdcmStaticAssert.h, 925
- GDCMImageIO2
 - itk::GDCMImageIO2, 380
- GDCMMACROENTRY_H
 - gdcmMacroEntry.h, 902
- gdcm, 103
 - AEComp, 117
 - ASComp, 117
 - backslash, 119
 - CSComp, 117
 - CompOperators, 117
 - DAComp, 117
 - DTComp, 117
 - eArabic, 118
 - eCyrillic, 118
 - eFind, 119
 - eGB18030, 118
 - eGreek, 118
 - eHebrew, 118
 - eImageOrFrame, 118
 - eJapanese, 118
 - eJapaneseKanjiMultibyte, 118
 - eJapaneseSupplementaryKanjiMultibyte, 118
 - eKoreanHangulHanjaMultibyte, 118
 - eLatin1, 118
 - eLatin2, 118
 - eLatin3, 118
 - eLatin4, 118
 - eLatin5, 118
 - eMove, 119
 - ePatient, 118
 - ePatientRootType, 119
 - eSeries, 118
 - eStudy, 118
 - eStudyRootType, 119
 - eThai, 118
 - eUTF8, 118
 - ECharSet, 118
 - EQueryLevel, 118
 - EQueryType, 118
 - ERootType, 119
 - FileList, 117
 - GDCM_DIFFERENT, 117
 - GDCM_EQUAL, 117
 - GDCM_GREATER, 118
 - GDCM_GREATEROREQUAL, 118
 - GDCM_LESS, 118
 - GDCM_LESSEQUAL, 118
 - GetVRFromTag, 119
 - GlobalInstance, 123
 - IconImage, 117
 - LD_ALL, 119
 - LD_NOSEQ, 119
 - LD_NOSHADOW, 119
 - LD_NOSHADOWSEQ, 119
 - LOComp, 117
 - LTComp, 117
 - LodModeType, 119
 - MacroEntry, 117
 - NestedMacroEntries, 117
 - operator<<, 119d
 - operator>>, 123
 - operator==, 123
 - PNComp, 117
 - SHComp, 117
 - STComp, 117
 - TMComp, 117
 - TYPETOENCODING, 123
 - to_string, 123
 - UIComp, 117
 - UTComp, 117
 - VRBINARY, 123
- gdcm2pnm.man, 865
- gdcm2vtk.man, 865
- gdcm::Attribute
 - VMType, 164
- gdcm::Attribute< Group, Element, TVR, VM::VM1 >
 - VMType, 170
- gdcm::CSAHeader
 - DATASET_FORMAT, 253
 - INTERFILE, 253
 - NOMAGIC, 253
 - SV10, 253
 - UNKNOWN, 253
 - ZEROED_OUT, 253
- gdcm::CryptographicMessageSyntax
 - AES128_CIPHER, 246
 - AES192_CIPHER, 246
 - AES256_CIPHER, 246
 - DES3_CIPHER, 246
 - DES_CIPHER, 246
- gdcm::DictConverter
 - DICT_DEBUG, 300
 - DICT_DEFAULT, 300

- DICT_XML, 300
- gdcmm::Dicts
 - GEMS, 306
 - PHILIPS, 306
 - SIEMENS, 306
- gdcmm::LookupTable
 - BLUE, 469
 - GRAY, 469
 - GREEN, 469
 - RED, 469
 - UNKNOWN, 469
- gdcmm::MediaStorage
 - AmbulatoryECGWaveformStorage, 480
 - Audio, 481
 - BasicTextSR, 480
 - BasicVoiceAudioWaveformStorage, 480
 - CSANonImageStorage, 480
 - CTImageStorage, 479
 - CardiacElectrophysiologyWaveformStorage, 480
 - ComprehensiveSR, 480
 - ComputedRadiographyImageStorage, 479
 - DetachedPatientManagementSOPClass, 480
 - DetachedStudyManagementSOPClass, 480
 - DetachedVisitManagementSOPClass, 480
 - DigitalIntraoralXRayImageStorageForProcessing, 479
 - DigitalIntraoralXrayImageStorageForPresentation, 479
 - DigitalMammographyImageStorageForPresentation, 479
 - DigitalMammographyImageStorageForProcessing, 479
 - DigitalXRayImageStorageForPresentation, 479
 - DigitalXRayImageStorageForProcessing, 479
 - EncapsulatedPDFStorage, 480
 - EnhancedCTImageStorage, 479
 - EnhancedMRIImageStorage, 479
 - EnhancedSR, 480
 - EnhancedXAImageStorage, 481
 - GEPrivate3DModelStorage, 480
 - GeneralECGWaveformStorage, 480
 - GeneralElectricMagneticResonanceImageStorage, 480
 - GrayscaleSoftcopyPresentationStateStorageSOP-Class, 480
 - HangingProtocolStorage, 480
 - HardcopyGrayscaleImageStorage, 480
 - HemodynamicWaveformStorage, 480
 - KeyObjectSelectionDocument, 480
 - LeadECGWaveformStorage, 480
 - MRIImageStorage, 479
 - MRSpectroscopyStorage, 479
 - MS_END, 481
 - MammographyCADSR, 480
 - MediaStorageDirectoryStorage, 479
 - ModalityPerformedProcedureStepSOPClass, 481
 - MultiframeGrayscaleByteSecondaryCaptureImageStorage, 479
 - MultiframeGrayscaleWordSecondaryCaptureImageStorage, 479
 - MultiframeSingleBitSecondaryCaptureImageStorage, 479
 - MultiframeTrueColorSecondaryCaptureImageStorage, 480
 - NoObject, 481
 - NuclearMedicineImageStorage, 480
 - NuclearMedicineImageStorageRetired, 479
 - ObjectEnd, 481
 - PDF, 481
 - PETImageStorage, 480
 - Philips3D, 480
 - PhilipsPrivateMRSyntheticImageStorage, 481
 - RTDoseStorage, 480
 - RTImageStorage, 480
 - RTIonBeamsTreatmentRecordStorage, 481
 - RTIonPlanStorage, 481
 - RTPlanStorage, 480
 - RTStructureSetStorage, 480
 - RawDataStorage, 480
 - SecondaryCaptureImageStorage, 479
 - Segmentation, 481
 - SegmentationStorage, 481
 - SpacialFiducialsStorage, 480
 - SpacialRegistrationStorage, 480
 - StandaloneCurveStorage, 480
 - StandaloneModalityLUTStorage, 480
 - StandaloneOverlayStorage, 480
 - StandaloneVOILUTStorage, 480
 - StudyComponentManagementSOPClass, 480
 - SurfaceSegmentationStorage, 481
 - ToshibaPrivateDataStorage, 480
 - URI, 481
 - UltrasoundImageStorage, 479
 - UltrasoundImageStorageRetired, 479
 - UltrasoundMultiFrameImageStorage, 479
 - UltrasoundMultiFrameImageStorageRetired, 479
 - VLPhotographicImageStorage, 481
 - VLWholeSlideMicroscopyImageStorage, 481
 - Video, 481
 - VideoEndoscopicImageStorage, 480
 - Waveform, 481
 - XRay3DAngiographicImageStorage, 481
 - XRayAngiographicBiPlaneImageStorageRetired, 480
 - XRayAngiographicImageStorage, 480
 - XRayRadiofluoroscopicImageStorage, 480
- gdcmm::MeshPrimitive
 - EDGE, 489
 - FACET, 489

- LINE, 489
- MPTType_END, 489
- TRIANGLE, 489
- TRIANGLE_FAN, 489
- TRIANGLE_STRIP, 489
- VERTEX, 489
- gdcmm::Orientation
 - AXIAL, 509
 - CORONAL, 509
 - OBLIQUE, 509
 - SAGITTAL, 509
 - UNKNOWN, 509
- gdcmm::Parser
 - DuplicateAttributeError, 518
 - JunkAfterDocElementError, 518
 - NoElementsError, 518
 - NoError, 518
 - NoMemoryError, 518
 - SyntaxError, 518
 - TagMismatchError, 518
 - UndefinedEntityError, 519
 - UnexpectedStateError, 519
- gdcmm::PhotometricInterpretation
 - ARGB, 532
 - CMYK, 532
 - HSV, 532
 - MONOCHROME1, 532
 - MONOCHROME2, 532
 - PALETTE_COLOR, 532
 - PI_END, 532
 - RGB, 532
 - UNKNOWN, 532
 - YBR_FULL, 532
 - YBR_FULL_422, 532
 - YBR_ICT, 532
 - YBR_PARTIAL_420, 532
 - YBR_PARTIAL_422, 532
 - YBR_RCT, 532
- gdcmm::PixelFormat
 - FLOAT16, 535
 - FLOAT32, 535
 - FLOAT64, 535
 - INT12, 535
 - INT16, 535
 - INT32, 535
 - INT8, 535
 - SINGLEBIT, 535
 - UINT12, 535
 - UINT16, 535
 - UINT32, 535
 - UINT8, 535
 - UNKNOWN, 535
- gdcmm::Printer
 - CONDENSED_STYLE, 564
 - VERBOSE_STYLE, 564
 - XML, 564
- gdcmm::STATIC_ASSERTION_FAILURE< true >
 - value, 652
- gdcmm::Segment
 - ALGOType_END, 606
 - AUTOMATIC, 606
 - MANUAL, 606
- gdcmm::Spacing
 - CALIBRATED, 648
 - DETECTOR, 648
 - MAGNIFIED, 648
 - UNKNOWN, 648
- gdcmm::Surface
 - NO, 673
 - POINTS, 674
 - STATES_END, 673
 - SURFACE, 674
 - UNKNOWN, 673
 - VIEWType_END, 674
 - WIREFRAME, 674
 - YES, 673
- gdcmm::SwapCode
 - BadBigEndian, 685
 - BadLittleEndian, 684
 - BigEndian, 684
 - LittleEndian, 684
 - Unknown, 684
- gdcmm::TransferSyntax
 - CT_private_ELE, 711
 - DeflatedExplicitVRLittleEndian, 710
 - Explicit, 710
 - ExplicitVRBigEndian, 710
 - ExplicitVRLittleEndian, 710
 - Implicit, 710
 - ImplicitVRBigEndianACRNEMA, 711
 - ImplicitVRBigEndianPrivateGE, 710
 - ImplicitVRLittleEndian, 710
 - JPEG2000, 711
 - JPEG2000Lossless, 711
 - JPEGBaselineProcess1, 710
 - JPEGExtendedProcess2_4, 710
 - JPEGExtendedProcess3_5, 710
 - JPEGFullProgressionProcess10_12, 710
 - JPEGLSLossless, 711
 - JPEGLSNearLossless, 711
 - JPEGLosslessProcess14, 711
 - JPEGLosslessProcess14_1, 711
 - JPEGSpectralSelectionProcess6_8, 710
 - JPIPRReferenced, 711
 - MPEG2MainProfile, 711
 - RLELossless, 711
 - TS_END, 711
 - Unknown, 710

- gdcmm::Type
 - T1, 716
 - T1C, 716
 - T2, 716
 - T2C, 716
 - T3, 716
 - UNKNOWN, 716
- gdcmm::UIDs
 - AmbulatoryECGWaveformStorage, 727
 - AudioSRStorageTrialRetired, 728
 - BasicAnnotationBoxSOPClass, 726
 - BasicColorImageBoxSOPClass, 726
 - BasicColorPrintManagementMetaSOPClass, 726
 - BasicFilmBoxSOPClass, 726
 - BasicFilmSessionSOPClass, 726
 - BasicGrayscaleImageBoxSOPClass, 726
 - BasicGrayscalePrintManagementMetaSOPClass, 726
 - BasicPrintImageOverlayBoxSOPClassRetired, 727
 - BasicStudyContentNotificationSOPClassRetired, 726
 - BasicTextSRStorage, 728
 - BasicVoiceAudioWaveformStorage, 727
 - BlendingSoftcopyPresentationStateStorageSOPClass, 728
 - BreastImagingRelevantPatientInformationQuery, 729
 - CTImageStorage, 727
 - CardiacElectrophysiologyWaveformStorage, 727
 - CardiacRelevantPatientInformationQuery, 730
 - ChestCADSRStorage, 729
 - ColorSoftcopyPresentationStateStorageSOPClass, 728
 - ComprehensiveSRStorage, 728
 - ComprehensiveSRStorageTrialRetired, 728
 - ComputedRadiographyImageStorage, 727
 - DICOMApplicationContextName, 726
 - DICOMControlledTerminology, 726
 - DICOMUIDRegistry, 726
 - DeflatedExplicitVRLittleEndian, 724
 - DeformableSpatialRegistrationStorage, 728
 - DetachedInterpretationManagementSOPClassRetired, 726
 - DetachedPatientManagementMetaSOPClassRetired, 726
 - DetachedPatientManagementSOPClassRetired, 726
 - DetachedResultsManagementMetaSOPClassRetired, 726
 - DetachedResultsManagementSOPClassRetired, 726
 - DetachedStudyManagementMetaSOPClassRetired, 726
 - DetachedStudyManagementSOPClassRetired, 726
 - DetachedVisitManagementSOPClassRetired, 726
 - DetailSRStorageTrialRetired, 728
 - dicomAETitle, 730
 - dicomApplicationCluster, 730
 - dicomAssociationAcceptor, 730
 - dicomAssociationInitiator, 730
 - dicomAuthorizedNodeCertificateReference, 730
 - dicomConfigurationRoot, 730
 - dicomDescription, 730
 - dicomDevice, 730
 - dicomDeviceName, 730
 - dicomDeviceSerialNumber, 730
 - dicomDevicesRoot, 730
 - dicomHostname, 730
 - dicomInstalled, 730
 - dicomInstitutionAddress, 730
 - dicomInstitutionDepartmentName, 730
 - dicomInstitutionName, 730
 - dicomIssuerOfPatientID, 730
 - dicomManufacturer, 730
 - dicomManufacturerModelName, 730
 - dicomNetworkAE, 730
 - dicomNetworkConnection, 731
 - dicomNetworkConnectionReference, 730
 - dicomPort, 730
 - dicomPreferredCalledAETitle, 730
 - dicomPreferredCallingAETitle, 730
 - dicomPrimaryDeviceType, 730
 - dicomRelatedDeviceReference, 730
 - dicomSOPClass, 730
 - dicomSoftwareVersion, 730
 - dicomStationName, 730
 - dicomSupportedCharacterSet, 730
 - dicomTLSCyphersuite, 730
 - dicomThisNodeCertificateReference, 730
 - dicomTransferCapability, 731
 - dicomTransferRole, 730
 - dicomTransferSyntax, 730
 - dicomUniqueAETitle, 731
 - dicomUniqueAETitlesRegistryRoot, 730
 - dicomVendorData, 730
 - DigitalIntraoralXRayImageStorageForPresentation, 727
 - DigitalIntraoralXRayImageStorageForProcessing, 727
 - DigitalMammographyXRayImageStorageForPresentation, 727
 - DigitalMammographyXRayImageStorageForProcessing, 727
 - DigitalXRayImageStorageForPresentation, 727
 - DigitalXRayImageStorageForProcessing, 727
 - EncapsulatedCDASStorage, 729
 - EncapsulatedPDFStorage, 729
 - EnhancedCTImageStorage, 727
 - EnhancedMRIImageStorage, 727
 - EnhancedSRStorage, 728

- EnhancedXAImageStorage, 728
- EnhancedXRFImageStorage, 728
- ExplicitVRBigEndian, 724
- ExplicitVRLittleEndian, 724
- GeneralECGWaveformStorage, 727
- GeneralPurposePerformedProcedureStepSOP-
Class, 729
- GeneralPurposeScheduledProcedureStepSOP-
Class, 729
- GeneralPurposeWorklistInformationModelFIND, 729
- GeneralPurposeWorklistManagementMetaSOP-
Class, 729
- GeneralRelevantPatientInformationQuery, 729
- GrayscaleSoftcopyPresentationStateStorageSOP-
Class, 728
- HangingProtocolInformationModelFIND, 730
- HangingProtocolInformationModelMOVE, 730
- HangingProtocolStorage, 730
- HardcopyColorImageStorageSOPClassRetired, 727
- HardcopyGrayscaleImageStorageSOPClassRetired,
727
- HemodynamicWaveformStorage, 727
- ICBM452T1FrameofReference, 726
- ICBMSingleSubjectMRIFrameofReference, 726
- ImageOverlayBoxSOPClassRetired, 727
- ImplicitVRLittleEndianDefaultTransferSyntaxforDIC-
OM, 724
- InstanceAvailabilityNotificationSOPClass, 729
- JPEG2000ImageCompression, 725
- JPEG2000ImageCompressionLosslessOnly, 725
- JPEG2000Part2MulticomponentImageCompression,
725
- JPEG2000Part2MulticomponentImageCompression-
LosslessOnly, 725
- JPEGBaselineProcess1DefaultTransferSyntaxfor-
LossyJPEG8BitImageCompression, 724
- JPEGExtendedHierarchicalProcess1618Retired, 725
- JPEGExtendedHierarchicalProcess1719Retired, 725
- JPEGExtendedProcess24DefaultTransferSyntaxfor-
LossyJPEG12BitImageCompressionProcess4only,
724
- JPEGExtendedProcess35Retired, 724
- JPEGFULLProgressionHierarchicalProcess2426-
Retired, 725
- JPEGFULLProgressionHierarchicalProcess2527-
Retired, 725
- JPEGFULLProgressionNonHierarchicalProcess1012-
Retired, 724
- JPEGFULLProgressionNonHierarchicalProcess1113-
Retired, 724
- JPEGLSLosslessImageCompression, 725
- JPEGLSLossyNearLosslessImageCompression, 725
- JPEGLosslessHierarchicalProcess28Retired, 725
- JPEGLosslessHierarchicalProcess29Retired, 725
- JPEGLosslessNonHierarchicalFirstOrderPrediction-
Process14SelectionValue1DefaultTransfer-
SyntaxforLosslessJPEGImageCompression,
725
- JPEGLosslessNonHierarchicalProcess14, 724
- JPEGLosslessNonHierarchicalProcess15Retired,
725
- JPEGSpectralSelectionHierarchicalProcess2022-
Retired, 725
- JPEGSpectralSelectionHierarchicalProcess2123-
Retired, 725
- JPEGSpectralSelectionNonHierarchicalProcess68-
Retired, 724
- JPEGSpectralSelectionNonHierarchicalProcess79-
Retired, 724
- JPIPRReferenced, 725
- JPIPRReferencedDeflate, 725
- KeyObjectSelectionDocumentStorage, 729
- MPEG2MainProfileMainLevel, 725
- MRImageStorage, 727
- MRSpectroscopyStorage, 727
- MammographyCADSRStorage, 728
- MediaCreationManagementSOPClassUID, 727
- MediaStorageDirectoryStorage, 725
- ModalityPerformedProcedureStepNotificationSOP-
Class, 726
- ModalityPerformedProcedureStepRetrieveSOP-
Class, 726
- ModalityPerformedProcedureStepSOPClass, 726
- ModalityWorklistInformationModelFIND, 729
- MultiframeGrayscaleByteSecondaryCaptureImage-
Storage, 727
- MultiframeGrayscaleWordSecondaryCaptureImage-
Storage, 727
- MultiframeSingleBitSecondaryCaptureImageStorage,
727
- MultiframeTrueColorSecondaryCaptureImage-
Storage, 727
- NuclearMedicineImageStorage, 728
- NuclearMedicineImageStorageRetired, 727
- OphthalmicPhotography16BitImageStorage, 728
- OphthalmicPhotography8BitImageStorage, 728
- OphthalmicTomographyImageStorage, 728
- PatientRootQueryRetrieveInformationModelFIND,
729
- PatientRootQueryRetrieveInformationModelGET,
729
- PatientRootQueryRetrieveInformationModelMOVE,
729
- PatientStudyOnlyQueryRetrieveInformationModelFI-
NDRetired, 729
- PatientStudyOnlyQueryRetrieveInformationModelIG-
ETRetired, 729

- PatientStudyOnlyQueryRetrieveInformationModelM-
 OVERetired, 729
 PositronEmissionTomographyImageStorage, 729
 PresentationLUTSOPClass, 727
 PrintJobSOPClass, 726
 PrintQueueManagementSOPClassRetired, 727
 PrintQueueSOPInstanceRetired, 727
 PrinterConfigurationRetrieveSOPClass, 726
 PrinterConfigurationRetrieveSOPInstance, 726
 PrinterSOPClass, 726
 PrinterSOPInstance, 726
 ProceduralEventLoggingSOPClass, 726
 ProceduralEventLoggingSOPInstance, 726
 ProcedureLogStorage, 728
 ProductCharacteristicsQuerySOPClass, 730
 PseudoColorSoftcopyPresentationStateStorageSO-
 PClass, 728
 PullPrintRequestSOPClassRetired, 727
 PullStoredPrintManagementMetaSOPClassRetired,
 727
 RFC2557MIMEencapsulation, 725
 RLELossless, 725
 RTBeamsDeliveryInstructionStorageSupplement74-
 FrozenDraft, 729
 RTBeamsTreatmentRecordStorage, 729
 RTBrachyTreatmentRecordStorage, 729
 RTConventionalMachineVerificationSupplement74-
 FrozenDraft, 729
 RTDoseStorage, 729
 RTImageStorage, 729
 RTIonBeamsTreatmentRecordStorage, 729
 RTIonMachineVerificationSupplement74FrozenDraft,
 729
 RTIonPlanStorage, 729
 RTPlanStorage, 729
 RTStructureSetStorage, 729
 RTTreatmentSummaryRecordStorage, 729
 RawDataStorage, 728
 RealWorldValueMappingStorage, 728
 ReferencedColorPrintManagementMetaSOPClass-
 Retired, 726
 ReferencedGrayscalePrintManagementMetaSOP-
 ClassRetired, 726
 ReferencedImageBoxSOPClassRetired, 726
 SPM2AVG152PDFFrameofReference, 725
 SPM2AVG152T1FrameofReference, 725
 SPM2AVG152T2FrameofReference, 725
 SPM2AVG305T1FrameofReference, 725
 SPM2BRAINMASKFrameofReference, 725
 SPM2CSFFFrameofReference, 725
 SPM2EPIFrameofReference, 725
 SPM2FILT1FrameofReference, 725
 SPM2GRAYFrameofReference, 725
 SPM2PDFFrameofReference, 725
 SPM2PETFrameofReference, 725
 SPM2SINGLESUBJT1FrameofReference, 725
 SPM2SPECTFrameofReference, 725
 SPM2T1FrameofReference, 725
 SPM2T2FrameofReference, 725
 SPM2TRANSMFrameofReference, 725
 SPM2WHITEFrameofReference, 725
 SecondaryCaptureImageStorage, 727
 SegmentationStorage, 728
 SpatialFiducialsStorage, 728
 SpatialRegistrationStorage, 728
 StandaloneCurveStorageRetired, 727
 StandaloneModalityLUTStorageRetired, 728
 StandaloneOverlayStorageRetired, 727
 StandalonePETCurveStorageRetired, 729
 StandaloneVOILUTStorageRetired, 728
 StereometricRelationshipStorage, 728
 StorageCommitmentPullModelSOPClassRetired,
 726
 StorageCommitmentPullModelSOPInstanceRetired,
 726
 StorageCommitmentPushModelSOPClass, 726
 StorageCommitmentPushModelSOPInstance, 726
 StorageServiceClass, 726
 StoredPrintStorageSOPClassRetired, 727
 StudyComponentManagementSOPClassRetired,
 726
 StudyRootQueryRetrieveInformationModelFIND, 729
 StudyRootQueryRetrieveInformationModelGET, 729
 StudyRootQueryRetrieveInformationModelMOVE,
 729
 SubstanceAdministrationLoggingSOPClass, 726
 SubstanceAdministrationLoggingSOPInstance, 726
 SubstanceApprovalQuerySOPClass, 730
 TalairachBrainAtlasFrameofReference, 725
 TextSRStorageTrialRetired, 728
 uid_1_2_840_10008_15_0_3_1, 736
 uid_1_2_840_10008_15_0_3_10, 736
 uid_1_2_840_10008_15_0_3_11, 736
 uid_1_2_840_10008_15_0_3_12, 736
 uid_1_2_840_10008_15_0_3_13, 736
 uid_1_2_840_10008_15_0_3_14, 736
 uid_1_2_840_10008_15_0_3_15, 737
 uid_1_2_840_10008_15_0_3_16, 737
 uid_1_2_840_10008_15_0_3_17, 737
 uid_1_2_840_10008_15_0_3_18, 737
 uid_1_2_840_10008_15_0_3_19, 737
 uid_1_2_840_10008_15_0_3_2, 736
 uid_1_2_840_10008_15_0_3_20, 737
 uid_1_2_840_10008_15_0_3_21, 737
 uid_1_2_840_10008_15_0_3_22, 737
 uid_1_2_840_10008_15_0_3_23, 737
 uid_1_2_840_10008_15_0_3_24, 737
 uid_1_2_840_10008_15_0_3_25, 737

uid_1_2_840_10008_15_0_3_26, 737
uid_1_2_840_10008_15_0_3_27, 737
uid_1_2_840_10008_15_0_3_28, 737
uid_1_2_840_10008_15_0_3_29, 737
uid_1_2_840_10008_15_0_3_3, 736
uid_1_2_840_10008_15_0_3_30, 737
uid_1_2_840_10008_15_0_3_31, 737
uid_1_2_840_10008_15_0_3_4, 736
uid_1_2_840_10008_15_0_3_5, 736
uid_1_2_840_10008_15_0_3_6, 736
uid_1_2_840_10008_15_0_3_7, 736
uid_1_2_840_10008_15_0_3_8, 736
uid_1_2_840_10008_15_0_3_9, 736
uid_1_2_840_10008_15_0_4_1, 737
uid_1_2_840_10008_15_0_4_2, 737
uid_1_2_840_10008_15_0_4_3, 737
uid_1_2_840_10008_15_0_4_4, 737
uid_1_2_840_10008_15_0_4_5, 737
uid_1_2_840_10008_15_0_4_6, 737
uid_1_2_840_10008_15_0_4_7, 737
uid_1_2_840_10008_15_0_4_8, 737
uid_1_2_840_10008_1_1, 731
uid_1_2_840_10008_1_2, 731
uid_1_2_840_10008_1_20_1, 732
uid_1_2_840_10008_1_20_1_1, 732
uid_1_2_840_10008_1_20_2, 732
uid_1_2_840_10008_1_20_2_1, 732
uid_1_2_840_10008_1_2_1, 731
uid_1_2_840_10008_1_2_1_99, 731
uid_1_2_840_10008_1_2_2, 731
uid_1_2_840_10008_1_2_4_100, 731
uid_1_2_840_10008_1_2_4_50, 731
uid_1_2_840_10008_1_2_4_51, 731
uid_1_2_840_10008_1_2_4_52, 731
uid_1_2_840_10008_1_2_4_53, 731
uid_1_2_840_10008_1_2_4_54, 731
uid_1_2_840_10008_1_2_4_55, 731
uid_1_2_840_10008_1_2_4_56, 731
uid_1_2_840_10008_1_2_4_57, 731
uid_1_2_840_10008_1_2_4_58, 731
uid_1_2_840_10008_1_2_4_59, 731
uid_1_2_840_10008_1_2_4_60, 731
uid_1_2_840_10008_1_2_4_61, 731
uid_1_2_840_10008_1_2_4_62, 731
uid_1_2_840_10008_1_2_4_63, 731
uid_1_2_840_10008_1_2_4_64, 731
uid_1_2_840_10008_1_2_4_65, 731
uid_1_2_840_10008_1_2_4_66, 731
uid_1_2_840_10008_1_2_4_70, 731
uid_1_2_840_10008_1_2_4_80, 731
uid_1_2_840_10008_1_2_4_81, 731
uid_1_2_840_10008_1_2_4_90, 731
uid_1_2_840_10008_1_2_4_91, 731
uid_1_2_840_10008_1_2_4_92, 731
uid_1_2_840_10008_1_2_4_93, 731
uid_1_2_840_10008_1_2_4_94, 731
uid_1_2_840_10008_1_2_4_95, 731
uid_1_2_840_10008_1_2_5, 731
uid_1_2_840_10008_1_2_6_1, 732
uid_1_2_840_10008_1_2_6_2, 732
uid_1_2_840_10008_1_3_10, 732
uid_1_2_840_10008_1_40, 732
uid_1_2_840_10008_1_40_1, 732
uid_1_2_840_10008_1_42, 732
uid_1_2_840_10008_1_42_1, 732
uid_1_2_840_10008_1_4_1_1, 732
uid_1_2_840_10008_1_4_1_10, 732
uid_1_2_840_10008_1_4_1_11, 732
uid_1_2_840_10008_1_4_1_12, 732
uid_1_2_840_10008_1_4_1_13, 732
uid_1_2_840_10008_1_4_1_14, 732
uid_1_2_840_10008_1_4_1_15, 732
uid_1_2_840_10008_1_4_1_16, 732
uid_1_2_840_10008_1_4_1_17, 732
uid_1_2_840_10008_1_4_1_18, 732
uid_1_2_840_10008_1_4_1_2, 732
uid_1_2_840_10008_1_4_1_3, 732
uid_1_2_840_10008_1_4_1_4, 732
uid_1_2_840_10008_1_4_1_5, 732
uid_1_2_840_10008_1_4_1_6, 732
uid_1_2_840_10008_1_4_1_7, 732
uid_1_2_840_10008_1_4_1_8, 732
uid_1_2_840_10008_1_4_1_9, 732
uid_1_2_840_10008_1_4_2_1, 732
uid_1_2_840_10008_1_4_2_2, 732
uid_1_2_840_10008_1_9, 732
uid_1_2_840_10008_2_16_4, 732
uid_1_2_840_10008_2_6_1, 732
uid_1_2_840_10008_3_1_1_1, 732
uid_1_2_840_10008_3_1_2_1_1, 732
uid_1_2_840_10008_3_1_2_1_4, 732
uid_1_2_840_10008_3_1_2_2_1, 732
uid_1_2_840_10008_3_1_2_3_1, 732
uid_1_2_840_10008_3_1_2_3_2, 732
uid_1_2_840_10008_3_1_2_3_3, 732
uid_1_2_840_10008_3_1_2_3_4, 732
uid_1_2_840_10008_3_1_2_3_5, 733
uid_1_2_840_10008_3_1_2_5_1, 733
uid_1_2_840_10008_3_1_2_5_4, 733
uid_1_2_840_10008_3_1_2_5_5, 733
uid_1_2_840_10008_3_1_2_6_1, 733
uid_1_2_840_10008_4_2, 733
uid_1_2_840_10008_5_1_1_1, 733
uid_1_2_840_10008_5_1_1_14, 733
uid_1_2_840_10008_5_1_1_15, 733
uid_1_2_840_10008_5_1_1_16, 733
uid_1_2_840_10008_5_1_1_16_376, 733
uid_1_2_840_10008_5_1_1_17, 733

uid_1_2_840_10008_5_1_1_17_376, 733
uid_1_2_840_10008_5_1_1_18, 733
uid_1_2_840_10008_5_1_1_18_1, 733
uid_1_2_840_10008_5_1_1_2, 733
uid_1_2_840_10008_5_1_1_22, 733
uid_1_2_840_10008_5_1_1_23, 733
uid_1_2_840_10008_5_1_1_24, 733
uid_1_2_840_10008_5_1_1_24_1, 733
uid_1_2_840_10008_5_1_1_25, 733
uid_1_2_840_10008_5_1_1_26, 733
uid_1_2_840_10008_5_1_1_27, 733
uid_1_2_840_10008_5_1_1_29, 733
uid_1_2_840_10008_5_1_1_30, 733
uid_1_2_840_10008_5_1_1_31, 733
uid_1_2_840_10008_5_1_1_32, 733
uid_1_2_840_10008_5_1_1_33, 733
uid_1_2_840_10008_5_1_1_4, 733
uid_1_2_840_10008_5_1_1_4_1, 733
uid_1_2_840_10008_5_1_1_4_2, 733
uid_1_2_840_10008_5_1_1_9, 733
uid_1_2_840_10008_5_1_1_9_1, 733
uid_1_2_840_10008_5_1_4_1_1_1, 733
uid_1_2_840_10008_5_1_4_1_1_10, 734
uid_1_2_840_10008_5_1_4_1_1_104_1, 735
uid_1_2_840_10008_5_1_4_1_1_104_2, 735
uid_1_2_840_10008_5_1_4_1_1_11, 734
uid_1_2_840_10008_5_1_4_1_1_11_1, 734
uid_1_2_840_10008_5_1_4_1_1_11_2, 734
uid_1_2_840_10008_5_1_4_1_1_11_3, 734
uid_1_2_840_10008_5_1_4_1_1_11_4, 734
uid_1_2_840_10008_5_1_4_1_1_128, 735
uid_1_2_840_10008_5_1_4_1_1_129, 735
uid_1_2_840_10008_5_1_4_1_1_12_1, 734
uid_1_2_840_10008_5_1_4_1_1_12_1_1, 734
uid_1_2_840_10008_5_1_4_1_1_12_2, 734
uid_1_2_840_10008_5_1_4_1_1_12_2_1, 734
uid_1_2_840_10008_5_1_4_1_1_12_3, 734
uid_1_2_840_10008_5_1_4_1_1_13_1_1, 734
uid_1_2_840_10008_5_1_4_1_1_13_1_2, 734
uid_1_2_840_10008_5_1_4_1_1_1_1, 733
uid_1_2_840_10008_5_1_4_1_1_1_1_1, 733
uid_1_2_840_10008_5_1_4_1_1_1_2, 733
uid_1_2_840_10008_5_1_4_1_1_1_2_1, 733
uid_1_2_840_10008_5_1_4_1_1_1_3, 733
uid_1_2_840_10008_5_1_4_1_1_1_3_1, 733
uid_1_2_840_10008_5_1_4_1_1_2, 733
uid_1_2_840_10008_5_1_4_1_1_20, 734
uid_1_2_840_10008_5_1_4_1_1_2_1, 733
uid_1_2_840_10008_5_1_4_1_1_3, 734
uid_1_2_840_10008_5_1_4_1_1_3_1, 734
uid_1_2_840_10008_5_1_4_1_1_4, 734
uid_1_2_840_10008_5_1_4_1_1_481_1, 735
uid_1_2_840_10008_5_1_4_1_1_481_2, 735
uid_1_2_840_10008_5_1_4_1_1_481_3, 735
uid_1_2_840_10008_5_1_4_1_1_481_4, 735
uid_1_2_840_10008_5_1_4_1_1_481_5, 735
uid_1_2_840_10008_5_1_4_1_1_481_6, 735
uid_1_2_840_10008_5_1_4_1_1_481_7, 735
uid_1_2_840_10008_5_1_4_1_1_481_8, 735
uid_1_2_840_10008_5_1_4_1_1_481_9, 735
uid_1_2_840_10008_5_1_4_1_1_4_1, 734
uid_1_2_840_10008_5_1_4_1_1_4_2, 734
uid_1_2_840_10008_5_1_4_1_1_5, 734
uid_1_2_840_10008_5_1_4_1_1_6, 734
uid_1_2_840_10008_5_1_4_1_1_66, 734
uid_1_2_840_10008_5_1_4_1_1_66_1, 734
uid_1_2_840_10008_5_1_4_1_1_66_2, 734
uid_1_2_840_10008_5_1_4_1_1_66_3, 734
uid_1_2_840_10008_5_1_4_1_1_66_4, 734
uid_1_2_840_10008_5_1_4_1_1_67, 734
uid_1_2_840_10008_5_1_4_1_1_6_1, 734
uid_1_2_840_10008_5_1_4_1_1_7, 734
uid_1_2_840_10008_5_1_4_1_1_77_1, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_1, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_2, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_3, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_4, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4, 735
uid_1_2_840_10008_5_1_4_1_1_77_1_6, 737
uid_1_2_840_10008_5_1_4_1_1_77_2, 735
uid_1_2_840_10008_5_1_4_1_1_7_1, 734
uid_1_2_840_10008_5_1_4_1_1_7_2, 734
uid_1_2_840_10008_5_1_4_1_1_7_3, 734
uid_1_2_840_10008_5_1_4_1_1_7_4, 734
uid_1_2_840_10008_5_1_4_1_1_8, 734
uid_1_2_840_10008_5_1_4_1_1_88_1, 735
uid_1_2_840_10008_5_1_4_1_1_88_11, 735
uid_1_2_840_10008_5_1_4_1_1_88_2, 735
uid_1_2_840_10008_5_1_4_1_1_88_22, 735
uid_1_2_840_10008_5_1_4_1_1_88_3, 735
uid_1_2_840_10008_5_1_4_1_1_88_33, 735
uid_1_2_840_10008_5_1_4_1_1_88_4, 735
uid_1_2_840_10008_5_1_4_1_1_88_40, 735
uid_1_2_840_10008_5_1_4_1_1_88_50, 735
uid_1_2_840_10008_5_1_4_1_1_88_59, 735
uid_1_2_840_10008_5_1_4_1_1_88_65, 735
uid_1_2_840_10008_5_1_4_1_1_88_67, 735
uid_1_2_840_10008_5_1_4_1_1_9, 734
uid_1_2_840_10008_5_1_4_1_1_9_1, 734
uid_1_2_840_10008_5_1_4_1_1_9_1_1, 734
uid_1_2_840_10008_5_1_4_1_1_9_1_2, 734
uid_1_2_840_10008_5_1_4_1_1_9_1_3, 734

- uid_1_2_840_10008_5_1_4_1_1_9_2_1, 734
- uid_1_2_840_10008_5_1_4_1_1_9_3_1, 734
- uid_1_2_840_10008_5_1_4_1_1_9_4_1, 734
- uid_1_2_840_10008_5_1_4_1_2_1_1, 735
- uid_1_2_840_10008_5_1_4_1_2_1_2, 735
- uid_1_2_840_10008_5_1_4_1_2_1_3, 735
- uid_1_2_840_10008_5_1_4_1_2_2_1, 735
- uid_1_2_840_10008_5_1_4_1_2_2_2, 736
- uid_1_2_840_10008_5_1_4_1_2_2_3, 736
- uid_1_2_840_10008_5_1_4_1_2_3_1, 736
- uid_1_2_840_10008_5_1_4_1_2_3_2, 736
- uid_1_2_840_10008_5_1_4_1_2_3_3, 736
- uid_1_2_840_10008_5_1_4_31, 736
- uid_1_2_840_10008_5_1_4_32, 736
- uid_1_2_840_10008_5_1_4_32_1, 736
- uid_1_2_840_10008_5_1_4_32_2, 736
- uid_1_2_840_10008_5_1_4_32_3, 736
- uid_1_2_840_10008_5_1_4_33, 736
- uid_1_2_840_10008_5_1_4_34_1, 736
- uid_1_2_840_10008_5_1_4_34_2, 736
- uid_1_2_840_10008_5_1_4_34_3, 736
- uid_1_2_840_10008_5_1_4_34_4, 736
- uid_1_2_840_10008_5_1_4_34_4_1, 736
- uid_1_2_840_10008_5_1_4_34_4_2, 736
- uid_1_2_840_10008_5_1_4_34_4_3, 736
- uid_1_2_840_10008_5_1_4_34_4_4, 736
- uid_1_2_840_10008_5_1_4_34_5, 736
- uid_1_2_840_10008_5_1_4_37_1, 736
- uid_1_2_840_10008_5_1_4_37_2, 736
- uid_1_2_840_10008_5_1_4_37_3, 736
- uid_1_2_840_10008_5_1_4_38_1, 736
- uid_1_2_840_10008_5_1_4_38_2, 736
- uid_1_2_840_10008_5_1_4_38_3, 736
- uid_1_2_840_10008_5_1_4_41, 736
- uid_1_2_840_10008_5_1_4_42, 736
- UltrasoundImageStorage, 727
- UltrasoundImageStorageRetired, 727
- UltrasoundMultiframeImageStorage, 727
- UltrasoundMultiframeImageStorageRetired, 727
- UnifiedProcedureStepEventSOPClass, 729
- UnifiedProcedureStepPullSOPClass, 729
- UnifiedProcedureStepPushSOPClass, 729
- UnifiedProcedureStepWatchSOPClass, 729
- UnifiedWorklistandProcedureStepSOPInstance, 729
- UnifiedWorklistandProcedureStepServiceClass, 729
- VLEndoscopicImageStorage, 728
- VImageStorageTrialRetired, 728
- VLMicroscopicImageStorage, 728
- VLMultiframeImageStorageTrialRetired, 728
- VLPotographicImageStorage, 728
- VLSlideCoordinatesMicroscopicImageStorage, 728
- VLWholeSlideMicroscopyImageStorage, 731
- VOILUTBoxSOPClass, 727
- VerificationSOPClass, 724
- VideoEndoscopicImageStorage, 728
- VideoMicroscopicImageStorage, 728
- VideoPhotographicImageStorage, 728
- WaveformStorageTrialRetired, 727
- XMLEncoding, 725
- XRay3DAngiographicImageStorage, 728
- XRay3DCraniofacialImageStorage, 728
- XRayAngiographicBiPlaneImageStorageRetired, 728
- XRayAngiographicImageStorage, 728
- XRayRadiationDoseSRStorage, 729
- XRayRadiofluoroscopicImageStorage, 728
- gdcmm::Usage
 - Conditional, 791
 - Invalid, 791
 - Mandatory, 791
 - UserOption, 791
- gdcmm::VM
 - VM0, 803
 - VM1, 803
 - VM10, 803
 - VM12, 803
 - VM16, 803
 - VM18, 803
 - VM1_2, 804
 - VM1_3, 804
 - VM1_32, 804
 - VM1_4, 804
 - VM1_5, 804
 - VM1_8, 804
 - VM1_99, 804
 - VM1_n, 804
 - VM2, 803
 - VM24, 803
 - VM256, 804
 - VM28, 803
 - VM2_2n, 804
 - VM2_n, 804
 - VM3, 803
 - VM30_30n, 804
 - VM32, 803
 - VM35, 803
 - VM3_3n, 804
 - VM3_4, 804
 - VM3_n, 804
 - VM4, 803
 - VM47_47n, 804
 - VM4_4n, 804
 - VM5, 803
 - VM6, 803
 - VM6_6n, 804
 - VM7_7n, 804
 - VM8, 803
 - VM9, 803
 - VM99, 804

- VM_END, 804
- gdcmm::VR
 - AE, 807
 - AS, 807
 - AT, 807
 - CS, 807
 - DA, 807
 - DS, 807
 - DT, 808
 - FD, 808
 - FL, 808
 - INVALID, 807
 - IS, 808
 - LO, 808
 - LT, 808
 - OB, 808
 - OB_OW, 808
 - OF, 808
 - OW, 808
 - PN, 808
 - SH, 808
 - SL, 808
 - SQ, 808
 - SS, 808
 - ST, 808
 - TM, 808
 - UI, 808
 - UL, 808
 - UN, 808
 - US, 808
 - US_SS, 808
 - US_SS_OW, 808
 - UT, 808
 - VL16, 808
 - VL32, 808
 - VR_END, 808
 - VR_VM1, 808
 - VRALL, 808
 - VRASCII, 808
 - VRBINARY, 808
- gdcmm::network
 - eAABORTPDUReceivedOpen, 128
 - eAABORTRequest, 128
 - eAASSOCIATE_RQPDUreceived, 128
 - eAASSOCIATERequestLocalUser, 128
 - eAASSOCIATEResponseAccept, 128
 - eAASSOCIATEResponseReject, 128
 - eARELEASE_RPPDUReceived, 128
 - eARELEASE_RQPDUReceivedOpen, 128
 - eARELEASERequest, 128
 - eARELEASEResponse, 128
 - eARTIMTimerExpired, 128
 - eASSOCIATE_ACPDUreceived, 128
 - eASSOCIATE_RJPDUreceived, 128
 - eEventDoesNotExist, 128
 - ePDATATFPDU, 128
 - ePDATArequest, 128
 - eSta10ReleaseCollisionAc, 128
 - eSta11ReleaseCollisionRq, 128
 - eSta12ReleaseCollisionAcLocal, 128
 - eSta13AwaitingClose, 128
 - eSta1Idle, 128
 - eSta2Open, 128
 - eSta3WaitLocalAssoc, 128
 - eSta4LocalAssocDone, 128
 - eSta5WaitRemoteAssoc, 128
 - eSta6TransferReady, 128
 - eSta7WaitRelease, 128
 - eSta8WaitLocalRelease, 128
 - eSta9ReleaseCollisionRqLocal, 128
 - eStaDoesNotExist, 128
 - eTransportConnConfirmLocal, 128
 - eTransportConnIndicLocal, 128
 - eTransportConnectionClosed, 128
 - eUnrecognizedPDUReceived, 128
- gdcmm::network::DIMSE
 - C_CANCEL_RQ, 309
 - C_ECHO_RQ, 309
 - C_ECHO_RSP, 309
 - C_FIND_RQ, 309
 - C_FIND_RSP, 309
 - C_GET_RQ, 309
 - C_GET_RSP, 309
 - C_MOVE_RQ, 309
 - C_MOVE_RSP, 309
 - C_STORE_RQ, 309
 - C_STORE_RSP, 309
 - N_ACTION_RQ, 309
 - N_ACTION_RSP, 309
 - N_CREATE_RQ, 309
 - N_CREATE_RSP, 309
 - N_DELETE_RQ, 309
 - N_DELETE_RSP, 309
 - N_EVENT_REPORT_RQ, 309
 - N_EVENT_REPORT_RSP, 309
 - N_GET_RQ, 309
 - N_GET_RSP, 309
 - N_SET_RQ, 309
 - N_SET_RSP, 309
- gdcmm::terminal
 - black, 130
 - blink, 130
 - blue, 130
 - bright, 130
 - CONSOLE, 130
 - cyan, 130
 - dim, 130
 - green, 130

- hidden, 130
- magenta, 130
- red, 130
- reset, 130
- reverse, 130
- underline, 130
- VT100, 130
- white, 130
- yellow, 130
- gdcmm::ASN1, 160
 - ~ASN1, 161
 - ASN1, 161
 - ParseDump, 161
 - ParseDumpFile, 161
 - TestPBKDF2, 161
- gdcmm::AbortEvent, 143
- gdcmm::AnonymizeEvent, 145
 - ~AnonymizeEvent, 146
 - AnonymizeEvent, 146
 - CheckEvent, 146
 - GetEventName, 146
 - GetTag, 146
 - MakeObject, 146
 - Self, 146
 - SetTag, 146
 - Superclass, 146
- gdcmm::Anonymizer, 147
 - ~Anonymizer, 150
 - Anonymizer, 150
 - BALCPPProtect, 150
 - BasicApplicationLevelConfidentialityProfile, 150
 - CanEmptyTag, 150
 - Empty, 150
 - GetBasicApplicationLevelConfidentialityProfile-Attributes, 150
 - GetCryptographicMessageSyntax, 150
 - GetFile, 150
 - New, 150
 - RecurseDataSet, 150
 - Remove, 151
 - RemoveGroupLength, 151
 - RemovePrivateTags, 151
 - RemoveRetired, 151
 - Replace, 151
 - SetCryptographicMessageSyntax, 151
 - SetFile, 151
- gdcmm::AnyEvent, 152
- gdcmm::ApplicationEntity, 154
 - Internal, 155
 - IsValid, 155
 - MaxLength, 155
 - MaxNumberOfComponents, 155
 - Padding, 155
 - Print, 155
 - Separator, 155
 - SetBlob, 155
 - Squeeze, 155
- gdcmm::Attribute
 - ArrayType, 164
 - GDCM_STATIC_ASSERT, 164
 - GetAsDataElement, 164
 - GetDictVM, 164
 - GetDictVR, 164
 - GetNumberOfValues, 165
 - GetTag, 165
 - GetVM, 166
 - GetVR, 166
 - GetValue, 165
 - GetValues, 165
 - Internal, 168
 - operator<, 166
 - operator==, 166
 - Print, 167
 - Set, 167
 - SetByteValue, 167
 - SetByteValueNoSwap, 167
 - SetFromDataElement, 167
 - SetFromDataSet, 168
 - SetValue, 168
 - SetValues, 168
- gdcmm::Attribute< Group, Element, TVR, TVM >, 162
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 169
 - ArrayType, 170
 - GetAsDataElement, 171
 - GetDictVM, 171
 - GetDictVR, 171
 - GetNumberOfValues, 171
 - GetTag, 171
 - GetVM, 171
 - GetVR, 172
 - GetValue, 171
 - GetValues, 171
 - Internal, 173
 - operator<, 172
 - operator==, 172
 - Print, 172
 - Set, 172
 - SetByteValue, 172
 - SetByteValueNoSwap, 172
 - SetFromDataElement, 172
 - SetFromDataSet, 173
 - SetValue, 173
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, 173
 - GetVM, 174
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, 175
 - GetVM, 176

- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 176
 - ~Attribute, 177
 - ArrayType, 177
 - Attribute, 177
 - GetAsDataElement, 178
 - GetDictVM, 178
 - GetDictVR, 178
 - GetNumberOfValues, 178
 - GetTag, 178
 - GetVM, 178
 - GetVR, 178
 - GetValue, 178
 - GetValues, 178
 - Print, 179
 - SetByteValue, 179
 - SetFromDataElement, 179
 - SetNumberOfValues, 179
 - SetValue, 179
 - SetValues, 180
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, 180
 - GetVM, 181
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, 181
 - GetVM, 183
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, 183
 - GetVM, 184
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, 184
 - GetVM, 186
- gdcmm::AudioCodec, 186
 - ~AudioCodec, 187
 - AudioCodec, 187
 - CanCode, 187
 - CanDecode, 187
 - Decode, 188
- gdcmm::Base64, 188
 - ~Base64, 188
 - Base64, 188
 - Decode, 188
 - Encode, 189
 - GetDecodeLength, 189
 - GetEncodeLength, 189
- gdcmm::BaseRootQuery, 193
 - ~BaseRootQuery, 195
 - AddQueryDataSet, 195
 - BaseRootQuery, 195
 - GetAbstractSyntaxUID, 195
 - GetQueryDataSet, 195
 - GetTagListByLevel, 195
 - InitializeDataSet, 195
 - mDataSet, 196
 - mHelpDescription, 196
 - mImage, 196
 - mPatient, 196
 - mRootType, 196
 - mSeries, 196
 - mStudy, 196
 - QueryFactory, 196
 - SetSearchParameter, 195
 - ValidateQuery, 195
 - WriteHelpFile, 196
 - WriteQuery, 196
- gdcmm::BasicOffsetTable, 199
 - BasicOffsetTable, 200
 - operator<<, 201
 - Read, 200
- gdcmm::Bitmap, 201
 - ~Bitmap, 204
 - AreOverlaysInPixelData, 204
 - Bitmap, 204
 - Clear, 204
 - ComputeLossyFlag, 204
 - Dimensions, 208
 - GetBuffer, 204
 - GetBuffer2, 204
 - GetBufferLength, 204
 - GetColumns, 205
 - GetDataElement, 205
 - GetDimension, 205
 - GetDimensions, 205
 - GetLUT, 205
 - GetNeedByteSwap, 205
 - GetNumberOfDimensions, 205
 - GetPhotometricInterpretation, 205
 - GetPixelFormat, 205, 206
 - GetPlanarConfiguration, 206
 - GetRows, 206
 - GetTransferSyntax, 206
 - ImageChangeTransferSyntax, 208
 - IsEmpty, 206
 - IsLossy, 206
 - IsTransferSyntaxCompatible, 206
 - LUT, 208
 - LUTPtr, 204
 - LossyFlag, 208
 - NeedByteSwap, 208
 - NumberOfDimensions, 208
 - PF, 208
 - PI, 208
 - PixelData, 208
 - PixmapReader, 208
 - PlanarConfiguration, 208
 - Print, 206
 - SetColumns, 206
 - SetDataElement, 206

- SetDimension, 206
- SetDimensions, 207
- SetLUT, 207
- SetLossyFlag, 207
- SetNeedByteSwap, 207
- SetNumberOfDimensions, 207
- SetPhotometricInterpretation, 207
- SetPixelFormat, 207
- SetPlanarConfiguration, 207
- SetRows, 207
- SetTransferSyntax, 208
- TS, 209
- TryJPEG2000Codec, 208
- TryJPEG2000Codec2, 208
- TryJPEGCodec, 208
- TryJPEGCodec2, 208
- TryJPEGLSCCodec, 208
- TryKAKADUCodec, 208
- TryPVRGCodec, 208
- TryRAWCodec, 208
- TryRLECodec, 208
- gdcmm::BitmapToBitmapFilter, 209
 - ~BitmapToBitmapFilter, 210
 - BitmapToBitmapFilter, 210
 - GetOutput, 211
 - Input, 211
 - Output, 211
 - SetInput, 211
- gdcmm::ByteBuffer, 211
 - ByteBuffer, 212
 - Get, 212
 - GetStart, 212
 - ShiftEnd, 212
 - UpdatePosition, 212
- gdcmm::ByteSwap
 - Swap, 212
 - SwapFromSwapCodeIntoSystem, 212
 - SwapRange, 213
 - SwapRangeFromSwapCodeIntoSystem, 213
 - SystemIsBigEndian, 213
 - SystemIsLittleEndian, 213
- gdcmm::ByteSwap< T >, 212
- gdcmm::ByteSwapFilter, 213
 - ~ByteSwapFilter, 214
 - ByteSwap, 214
 - ByteSwapFilter, 214
 - SetByteSwapTag, 214
- gdcmm::ByteValue, 214
 - ~ByteValue, 216
 - ByteValue, 216
 - Clear, 216
 - Fill, 216
 - GetBuffer, 216
 - GetLength, 216
 - GetPointer, 217
 - IsEmpty, 217
 - IsPrintable, 217
 - operator const std::vector< char > &, 217
 - operator=, 217
 - operator==, 217
 - Print, 217
 - PrintASCII, 217
 - PrintGroupLength, 218
 - PrintHex, 218
 - Read, 218
 - SetLength, 218
 - Write, 218
 - WriteBuffer, 218
- gdcmm::CP246ExplicitDataElement, 243
 - GetLength, 245
 - Read, 245
 - ReadPreValue, 245
 - ReadValue, 245
 - ReadWithLength, 245
- gdcmm::CSAElement, 247
 - CSAElement, 249
 - DataField, 251
 - DataPtr, 249
 - GetByteValue, 249
 - GetKey, 249
 - GetName, 249
 - GetNoOfItems, 249
 - GetSyngoDT, 249
 - GetVM, 250
 - GetVR, 250
 - GetValue, 249
 - IsEmpty, 250
 - KeyField, 251
 - NameField, 251
 - NoOfItemsField, 251
 - operator<, 250
 - operator<<, 251
 - operator=, 250
 - operator==, 250
 - SetByteValue, 250
 - SetKey, 250
 - SetName, 250
 - SetNoOfItems, 250
 - SetSyngoDT, 250
 - SetVM, 250
 - SetVR, 250
 - SetValue, 250
 - SyngoDTField, 251
 - VRField, 251
 - ValueMultiplicityField, 251
- gdcmm::CSAHeader, 251
 - ~CSAHeader, 253
 - CSAHeader, 253

- CSAHeaderType, 253
- FindCSAElementByName, 253
- GetCSADatInfo, 254
- GetCSAEEnd, 254
- GetCSAElementByName, 254
- GetCSAImageHeaderInfoTag, 254
- GetCSASeriesHeaderInfoTag, 254
- GetDataSet, 254
- GetFormat, 255
- GetInterfile, 255
- LoadFromDataElement, 255
- operator<<, 255
- Print, 255
- Read, 255
- Write, 255
- gdcmm::CSAHeaderDict, 255
 - AddCSAHeaderDictEntry, 256
 - Begin, 257
 - CSAHeaderDict, 256
 - ConstIterator, 256
 - Dicts, 257
 - End, 257
 - GetCSAHeaderDictEntry, 257
 - IsEmpty, 257
 - Iterator, 256
 - LoadDefault, 257
 - MapCSAHeaderDictEntry, 256
 - operator<<, 257
- gdcmm::CSAHeaderDictEntry, 257
 - CSAHeaderDictEntry, 258
 - GetDescription, 258
 - GetName, 258
 - GetVM, 258
 - GetVR, 258
 - operator<, 258
 - operator<<, 259
 - SetDescription, 259
 - SetName, 259
 - SetVM, 259
 - SetVR, 259
- gdcmm::CSAHeaderDictException, 259
- gdcmm::CodeString, 232
 - CodeString, 233
 - const_iterator, 233
 - const_reference, 233
 - const_reverse_iterator, 233
 - difference_type, 233
 - GetAsString, 234
 - IsValid, 234
 - iterator, 233
 - operator<<, 234
 - operator==, 234
 - pointer, 233
 - reference, 233
 - reverse_iterator, 233
 - Size, 234
 - size_type, 233
 - TrimInternal, 234
 - value_type, 233
- gdcmm::Codec, 229
- gdcmm::Coder, 230
 - ~Coder, 231
 - CanCode, 231
 - Code, 231
 - InternalCode, 231
- gdcmm::Command, 234
 - ~Command, 236
 - Command, 236
 - Execute, 236
- gdcmm::CommandDataSet, 236
 - ~CommandDataSet, 238
 - CommandDataSet, 238
 - Insert, 238
 - operator<<, 238
 - Read, 238
 - Replace, 238
 - Write, 238
- gdcmm::CompositeNetworkFunctions, 240
 - CEcho, 241
 - CFind, 241
 - CMove, 241
 - CStore, 242
 - ConstructQuery, 242
 - KeyValuePairArrayType, 241
 - KeyValuePairType, 241
- gdcmm::ConstCharWrapper, 243
 - ConstCharWrapper, 243
 - operator const char *, 243
- gdcmm::CryptographicMessageSyntax, 245
 - ~CryptographicMessageSyntax, 246
 - CipherTypes, 246
 - CryptographicMessageSyntax, 246
 - Decrypt, 246
 - Encrypt, 246
 - GetCipherType, 246
 - ParseCertificateFile, 247
 - ParseKeyFile, 247
 - SetCipherType, 247
- gdcmm::Curve, 263
 - ~Curve, 264
 - Curve, 264
 - Decode, 264
 - GetAsPoints, 264
 - GetDataValueRepresentation, 264
 - GetDimensions, 265
 - GetGroup, 265
 - GetNumberOfCurves, 265
 - GetNumberOfPoints, 265

- GetTypeInfoData, 265
- GetTypeInfoDataDescription, 265
- IsEmpty, 265
- Print, 265
- SetCoordinateStartValue, 265
- SetCoordinateStepValue, 265
- SetCurve, 265
- SetCurveDataDescriptor, 265
- SetCurveDescription, 265
- SetDataValueRepresentation, 265
- SetDimensions, 265
- SetGroup, 265
- SetNumberOfPoints, 265
- SetTypeInfoData, 265
- Update, 265
- gdcm::DICOMDIR, 293
 - DICOMDIR, 294
- gdcm::DICOMDIRGenerator, 294
 - ~DICOMDIRGenerator, 295
 - AddImageDirectoryRecord, 295
 - AddPatientDirectoryRecord, 295
 - AddSeriesDirectoryRecord, 295
 - AddStudyDirectoryRecord, 295
 - DICOMDIRGenerator, 295
 - FilenameType, 295
 - FileNamesType, 295
 - Generate, 295
 - GetFile, 295
 - GetScanner, 296
 - SetDescriptor, 296
 - SetFile, 296
 - SetFileNames, 296
 - SetRootDirectory, 296
- gdcm::DataElement, 265
 - Clear, 269
 - DataElement, 268
 - Empty, 269
 - GetByteValue, 269
 - GetLength, 269
 - GetSequenceOfFragments, 269
 - GetSequenceOfItems, 269, 270
 - GetTag, 270
 - GetVL, 271
 - GetVR, 271
 - GetValue, 270
 - GetValueAsSQ, 270
 - IsEmpty, 271
 - IsUndefinedLength, 271
 - operator<, 271
 - operator<<, 274
 - operator=, 272
 - operator==, 272
 - Read, 272
 - ReadOrSkip, 272
 - ReadPreValue, 272
 - ReadValue, 272
 - ReadWithLength, 272
 - SetByteValue, 272
 - SetTag, 273
 - SetVL, 273
 - SetVLToUndefined, 273
 - SetVR, 273
 - SetValue, 273
 - TagField, 274
 - VRField, 274
 - ValueField, 274
 - ValueLengthField, 274
 - ValuePtr, 268
 - Write, 274
- gdcm::DataElementException, 275
- gdcm::DataEvent, 275
 - ~DataEvent, 277
 - CheckEvent, 277
 - DataEvent, 277
 - GetData, 277
 - GetDataLength, 277
 - GetEventName, 277
 - MakeObject, 277
 - Self, 277
 - SetData, 277
 - Superclass, 277
- gdcm::DataSet, 278
 - Begin, 280
 - CSAHeader, 284
 - Clear, 280
 - ComputeDataElement, 280
 - ComputeGroupLength, 281
 - ConstIterator, 280
 - DataElementSet, 280
 - End, 281
 - FindDataElement, 281
 - FindNextDataElement, 281
 - GetDEEnd, 282
 - GetDES, 282
 - GetDataElement, 281, 282
 - GetLength, 282
 - GetPrivateCreator, 282
 - Insert, 282
 - InsertDataElement, 282
 - IsEmpty, 282
 - Iterator, 280
 - operator<<, 284
 - operator(), 283
 - operator=, 283
 - Print, 283
 - Read, 283
 - ReadNested, 283
 - ReadSelectedTags, 283

- ReadSelectedTagsWithLength, 283
- ReadUpToTag, 283
- ReadUpToTagWithLength, 283
- ReadWithLength, 283
- Remove, 283
- Replace, 283
- ReplaceEmpty, 284
- Size, 284
- SizeType, 280
- Write, 284
- gdcmm::DataSetEvent, 284
 - ~DataSetEvent, 286
 - CheckEvent, 286
 - DataSetEvent, 286
 - GetDataSet, 286
 - GetEventName, 286
 - MakeObject, 286
 - Self, 286
 - Superclass, 286
- gdcmm::DataSetHelper, 287
 - ComputeVR, 287
- gdcmm::Decoder, 287
 - ~Decoder, 288
 - CanDecode, 288
 - Decode, 288
- gdcmm::DefinedTerms, 289
 - DefinedTerms, 289
- gdcmm::Defs, 289
 - ~Defs, 290
 - Defs, 290
 - GetIODFromFile, 290
 - GetIODNameFromMediaStorage, 290
 - GetIODs, 290, 291
 - GetMacros, 291
 - GetModules, 291
 - GetTypeFromTag, 291
 - Global, 291
 - IsEmpty, 291
 - LoadDefaults, 291
 - LoadFromFile, 291
 - Verify, 291
- gdcmm::DeltaEncodingCodec, 291
 - ~DeltaEncodingCodec, 293
 - CanDecode, 293
 - Decode, 293
 - DeltaEncodingCodec, 293
- gdcmm::Dict, 296
 - AddDictEntry, 297
 - Begin, 297
 - ConstIterator, 297
 - Dict, 297
 - Dicts, 298
 - End, 297
 - GetDictEntry, 298
 - GetDictEntryByKeyword, 298
 - GetDictEntryByName, 298
 - GetKeywordFromTag, 298
 - IsEmpty, 298
 - Iterator, 297
 - LoadDefault, 298
 - MapDictEntry, 297
 - operator<<, 298
- gdcmm::DictConverter, 299
 - ~DictConverter, 300
 - AddGroupLength, 300
 - Convert, 300
 - ConvertToCXX, 300
 - ConvertToXML, 300
 - DictConverter, 300
 - GetDictName, 300
 - GetInputFilename, 300
 - GetOutputFilename, 300
 - GetOutputType, 300
 - OutputTypes, 300
 - ReadVM, 300
 - ReadVR, 300
 - Readuint16, 300
 - SetDictName, 300
 - SetInputFileName, 300
 - SetOutputFileName, 300
 - SetOutputType, 300
 - WriteFooter, 300
 - WriteHeader, 301
- gdcmm::DictEntry, 301
 - DictEntry, 302
 - GetKeyword, 302
 - GetName, 302
 - GetRetired, 302
 - GetVM, 302
 - GetVR, 302
 - IsUnique, 303
 - operator<<, 303
 - SetElementXX, 303
 - SetGroupXX, 303
 - SetKeyword, 303
 - SetName, 303
 - SetRetired, 303
 - SetVM, 303
 - SetVR, 303
- gdcmm::DictPrinter, 303
 - ~DictPrinter, 305
 - DictPrinter, 305
 - Print, 305
 - PrintDataElement2, 305
 - PrintDataSet2, 305
- gdcmm::Dicts, 305
 - ~Dicts, 307
 - ConstructorType, 306

- Dicts, 307
- GetCSAHeaderDict, 307
- GetConstructorString, 307
- GetDictEntry, 307
- GetPrivateDict, 307
- GetPublicDict, 307
- Global, 307
- IsEmpty, 307
- LoadDefaults, 307
- operator<<, 307
- gdcmm::DirectionCosines, 309
 - ~DirectionCosines, 310
 - ComputeDistAlongNormal, 310
 - Cross, 310
 - CrossDot, 310
 - DirectionCosines, 310
 - Dot, 310
 - IsValid, 311
 - Normalize, 311
 - operator const double *, 311
 - Print, 311
 - SetFromString, 311
- gdcmm::Directory, 311
 - ~Directory, 313
 - Directory, 313
 - Explore, 313
 - FilenameType, 312
 - FileNamesType, 312
 - GetDirectories, 313
 - GetFileNames, 313
 - GetToplevel, 313
 - Load, 313
 - operator<<, 314
 - Print, 313
- gdcmm::DirectoryHelper, 314
 - GetCTImageSeriesUIDs, 314
 - GetFileNamesFromSeriesUIDs, 314
 - GetFrameOfReference, 314
 - GetMRIImageSeriesUIDs, 315
 - GetRTStructSeriesUIDs, 315
 - GetSOPClassUID, 315
 - GetSeriesUIDsBySOPClassUID, 315
 - LoadImageFromFiles, 315
 - RetrieveSOPInstanceUIDFromIndex, 315
 - RetrieveSOPInstanceUIDFromZPosition, 315
- gdcmm::DummyValueGenerator, 315
 - Generate, 316
- gdcmm::Dumper, 316
 - ~Dumper, 317
 - Dumper, 317
- gdcmm::Element
 - GetAsDataElement, 319
 - GetLength, 319
 - GetVM, 320
 - GetVR, 320
 - GetValue, 319
 - GetValues, 320
 - Internal, 320
 - Print, 320
 - Read, 320
 - Set, 320
 - SetFromDataElement, 320
 - SetNoSwap, 320
 - SetValue, 320
 - Type, 319
 - Write, 320
- gdcmm::Element< TVR, TVM >, 318
- gdcmm::Element< TVR, VM::VM1_2 >, 321
 - Parent, 322
 - SetLength, 322
- gdcmm::Element< TVR, VM::VM1_n >, 322
 - ~Element, 324
 - Element, 324
 - GetAsDataElement, 324
 - GetLength, 324
 - GetVM, 324
 - GetVR, 324
 - GetValue, 324
 - operator=, 324
 - Print, 324
 - Read, 324
 - Set, 325
 - SetArray, 325
 - SetFromDataElement, 325
 - SetLength, 325
 - SetNoSwap, 325
 - SetValue, 325
 - Type, 324
 - Write, 325
 - WriteASCII, 325
- gdcmm::Element< TVR, VM::VM2_2n >, 325
 - Parent, 327
 - SetLength, 327
- gdcmm::Element< TVR, VM::VM2_n >, 327
 - Parent, 329
 - SetLength, 329
- gdcmm::Element< TVR, VM::VM3_3n >, 329
 - Parent, 330
 - SetLength, 330
- gdcmm::Element< TVR, VM::VM3_n >, 331
 - Parent, 332
 - SetLength, 332
- gdcmm::Element< VR::AS, VM::VM5 >, 332
 - GetLength, 332
 - Internal, 332
 - Print, 332
- gdcmm::Element< VR::OB, VM::VM1 >, 333
- gdcmm::Element< VR::OW, VM::VM1 >, 334

- gdcmm::EncapsulatedDocument, 336
 - EncapsulatedDocument, 336
- gdcmm::EncodingImplementation< VR::VRASCII >, 336
 - Read, 337
 - ReadComputeLength, 337
 - ReadNoSwap, 337
 - Write, 337
- gdcmm::EncodingImplementation< VR::VRBINARY >, 337
 - Read, 338
 - ReadComputeLength, 338
 - ReadNoSwap, 338
 - Write, 338
- gdcmm::EndEvent, 338
- gdcmm::EnumeratedValues, 339
 - EnumeratedValues, 340
- gdcmm::Event, 340
 - ~Event, 341
 - CheckEvent, 341
 - Event, 341
 - GetEventName, 341
 - MakeObject, 341
 - Print, 341
- gdcmm::Exception, 342
 - ~Exception, 343
 - Exception, 343
 - GetDescription, 343
 - what, 343
- gdcmm::ExitEvent, 343
- gdcmm::ExplicitDataElement, 344
 - GetLength, 346
 - Read, 346
 - ReadPreValue, 346
 - ReadValue, 346
 - ReadWithLength, 346
 - Write, 346
- gdcmm::ExplicitImplicitDataElement, 346
 - GetLength, 348
 - Read, 348
 - ReadPreValue, 348
 - ReadValue, 348
 - ReadWithLength, 348
- gdcmm::Fiducials, 348
 - Fiducials, 349
- gdcmm::File, 349
 - ~File, 351
 - File, 351
 - GetDataSet, 351
 - GetHeader, 351
 - operator<<, 352
 - Read, 352
 - SetDataSet, 352
 - SetHeader, 352
 - Write, 352
- gdcmm::FileDerivation, 352
 - ~FileDerivation, 353
 - AddDerivationDescription, 353
 - AddPurposeOfReferenceCodeSequence, 353
 - AddReference, 353
 - AddSourceImageSequence, 353
 - Derive, 354
 - FileDerivation, 353
 - GetFile, 354
 - SetDerivationCodeSequenceCodeValue, 354
 - SetDerivationDescription, 354
 - SetFile, 354
 - SetPurposeOfReferenceCodeSequenceCodeValue, 354
- gdcmm::FileExplicitFilter, 355
 - ~FileExplicitFilter, 356
 - Change, 356
 - ChangeFMI, 356
 - FileExplicitFilter, 356
 - GetFile, 356
 - ProcessDataSet, 356
 - SetChangePrivateTags, 356
 - SetFile, 356
 - SetRecomputeItemLength, 356
 - SetRecomputeSequenceLength, 356
 - SetUseVRUN, 356
- gdcmm::FileMetaInformation, 357
 - ~FileMetaInformation, 359
 - AppendImplementationClassUID, 359
 - ComputeDataSetMediaStorageSOPClass, 359
 - ComputeDataSetTransferSyntax, 359
 - DataSetMS, 362
 - DataSetTS, 362
 - Default, 359
 - FileMetaInformation, 359
 - FillFromDataSet, 359
 - GetDataSetTransferSyntax, 359
 - GetFileMetaInformationVersion, 360
 - GetFullLength, 360
 - GetGDCMImplementationClassUID, 360
 - GetGDCMImplementationVersionName, 360
 - GetGDCMSourceApplicationEntityTitle, 360
 - GetImplementationClassUID, 360
 - GetImplementationVersionName, 360
 - GetMediaStorage, 360
 - GetMetaInformationTS, 360
 - GetPreamble, 360
 - GetSourceApplicationEntityTitle, 360
 - Insert, 360
 - IsValid, 360
 - MetaInformationTS, 362
 - operator<<, 362
 - Read, 361
 - ReadCompat, 361
 - ReadCompatInternal, 361

- Replace, 361
- SetDataSetTransferSyntax, 361
- SetImplementationClassUID, 361
- SetImplementationVersionName, 361
- SetPreamble, 361
- SetSourceApplicationEntityTitle, 361
- Write, 361
- gdcM::FileSet, 366
 - AddFile, 367
 - FileSet, 367
 - FileType, 367
 - FilesType, 367
 - GetFiles, 367
 - operator<<, 367
 - SetFiles, 367
- gdcM::FileWithName, 368
 - FileWithName, 369
 - filename, 369
- gdcM::Filename, 362
 - Filename, 363
 - GetExtension, 363
 - GetFileName, 363
 - GetName, 363
 - GetPath, 363
 - IsEmpty, 363
 - IsIdentical, 363
 - Join, 363
 - operator const char *, 363
 - ToUnixSlashes, 363
 - ToWindowsSlashes, 364
- gdcM::FilenameGenerator, 364
 - ~FilenameGenerator, 365
 - FilenameGenerator, 365
 - FilenameType, 365
 - FilenamesType, 365
 - Generate, 365
 - GetFilename, 365
 - GetFilenames, 365
 - GetNumberOfFilenames, 365
 - GetPattern, 366
 - GetPrefix, 366
 - SetNumberOfFilenames, 366
 - SetPattern, 366
 - SetPrefix, 366
 - SizeType, 365
- gdcM::FindPatientRootQuery, 369
 - FindPatientRootQuery, 371
 - GetAbstractSyntaxUID, 371
 - GetTagListByLevel, 371
 - InitializeDataSet, 371
 - QueryFactory, 371
 - ValidateQuery, 371
- gdcM::FindStudyRootQuery, 372
 - FindStudyRootQuery, 373
 - GetAbstractSyntaxUID, 373
 - GetTagListByLevel, 373
 - InitializeDataSet, 373
 - QueryFactory, 373
 - ValidateQuery, 373
- gdcM::Fragment, 374
 - Fragment, 376
 - GetLength, 376
 - operator<<, 376
 - Read, 376
 - ReadValue, 376
 - Write, 376
- gdcM::Global, 384
 - ~Global, 385
 - Append, 385
 - GetDefs, 385
 - GetDicts, 385
 - GetInstance, 385
 - Global, 385
 - LoadResourcesFiles, 385
 - Locate, 386
 - operator<<, 386
 - Prepend, 386
- gdcM::GroupDict, 386
 - ~GroupDict, 387
 - Add, 387
 - GetAbbreviation, 387
 - GetName, 387
 - GroupDict, 387
 - GroupStringVector, 387
 - Insert, 387
 - operator<<, 388
 - Size, 387
- gdcM::IOD, 435
 - AddIODEntry, 436
 - Clear, 436
 - GetIODEntry, 436
 - GetNumberOfIODs, 436
 - GetTypeFromTag, 436
 - IOD, 435
 - MapIODEntry, 435
 - operator<<, 436
 - SizeType, 435
- gdcM::IODEntry, 436
 - GetIE, 437
 - GetName, 437
 - GetRef, 437
 - GetUsage, 437
 - GetUsageType, 438
 - IODEntry, 437
 - operator<<, 438
 - SetIE, 438
 - SetName, 438
 - SetRef, 438

- SetUsage, 438
- gdcm::IODs, 438
 - AddIOD, 439
 - Begin, 439
 - Clear, 439
 - End, 439
 - GetIOD, 439
 - IODMapType, 439
 - IODMapTypeConstIterator, 439
 - IODName, 439
 - IODs, 439
 - operator<<, 439
- gdcm::IPPSorter, 439
 - ~IPPSorter, 441
 - ComputeZSpacing, 443
 - DirCosTolerance, 443
 - GetDirectionCosinesTolerance, 441
 - GetZSpacing, 441
 - GetZSpacingTolerance, 442
 - IPPSorter, 441
 - SetComputeZSpacing, 442
 - SetDirectionCosinesTolerance, 442
 - SetZSpacingTolerance, 442
 - Sort, 442
 - ZSpacing, 443
 - ZTolerance, 443
- gdcm::IconImageFilter, 388
 - ~IconImageFilter, 389
 - Extract, 389
 - ExtractIconImages, 389
 - ExtractVeprolIconImages, 389
 - GetFile, 389
 - GetIconImage, 389
 - GetNumberOfIconImages, 389
 - IconImageFilter, 389
 - SetFile, 390
- gdcm::IconImageGenerator, 390
 - ~IconImageGenerator, 391
 - AutoPixelMinMax, 391
 - ConvertRGBToPaletteColor, 391
 - Generate, 391
 - GetIconImage, 391
 - GetPixmap, 392
 - IconImageGenerator, 391
 - SetOutputDimensions, 392
 - SetOutsideValuePixel, 392
 - SetPixelMinMax, 392
 - SetPixmap, 392
- gdcm::Image, 393
 - ~Image, 395
 - GetDirectionCosines, 395
 - GetIntercept, 395
 - GetOrigin, 395
 - GetSlope, 396
 - GetSpacing, 396
 - Image, 395
 - Print, 396
 - SetDirectionCosines, 396
 - SetIntercept, 396
 - SetOrigin, 396
 - SetSlope, 396
 - SetSpacing, 396
- gdcm::ImageApplyLookupTable, 397
 - ~ImageApplyLookupTable, 399
 - Apply, 399
 - ImageApplyLookupTable, 399
- gdcm::ImageChangePhotometricInterpretation, 399
 - ~ImageChangePhotometricInterpretation, 402
 - Change, 402
 - ChangeMonochrome, 402
 - GetPhotometricInterpretation, 402
 - ImageChangePhotometricInterpretation, 402
 - RGB2YBR, 402
 - SetPhotometricInterpretation, 402
 - YBR2RGB, 402
- gdcm::ImageChangePlanarConfiguration, 403
 - ~ImageChangePlanarConfiguration, 405
 - Change, 405
 - GetPlanarConfiguration, 405
 - ImageChangePlanarConfiguration, 405
 - RGBPixelsToRGBPlanes, 405
 - RGBPlanesToRGBPixels, 405
 - SetPlanarConfiguration, 405
- gdcm::ImageChangeTransferSyntax, 406
 - ~ImageChangeTransferSyntax, 408
 - Change, 408
 - GetTransferSyntax, 408
 - ImageChangeTransferSyntax, 408
 - SetCompressIconImage, 408
 - SetForce, 409
 - SetTransferSyntax, 409
 - SetUserCodec, 409
 - TryJPEG2000Codec, 409
 - TryJPEGCodec, 409
 - TryJPEGLSCodec, 409
 - TryRAWCodec, 409
 - TryRLECodec, 409
- gdcm::ImageCodec, 410
 - ~ImageCodec, 412
 - CanCode, 412
 - CanDecode, 412
 - Decode, 412
 - Dimensions, 414
 - DoByteSwap, 413
 - DoInvertMonochrome, 413
 - DoOverlayCleanup, 413
 - DoPaddedCompositePixelCode, 413
 - DoPlanarConfiguration, 413

- DoSimpleCopy, 413
- DoYBR, 413
- GetDimensions, 413
- GetHeaderInfo, 413
- GetLUT, 413
- GetLossyFlag, 413
- GetNeedByteSwap, 413
- GetNumberOfDimensions, 413
- GetPhotometricInterpretation, 413
- GetPixelFormat, 413
- GetPlanarConfiguration, 413
- ImageChangePhotometricInterpretation, 414
- ImageCodec, 412
- IsLossy, 413
- IsValid, 413
- LUT, 415
- LUTPtr, 412
- LossyFlag, 415
- NeedByteSwap, 415
- NeedOverlayCleanup, 415
- NumberOfDimensions, 415
- PF, 415
- PI, 415
- PlanarConfiguration, 415
- RequestPaddedCompositePixelCode, 415
- RequestPlanarConfiguration, 415
- SetDimensions, 414
- SetLUT, 414
- SetLossyFlag, 414
- SetNeedByteSwap, 414
- SetNeedOverlayCleanup, 414
- SetNumberOfDimensions, 414
- SetPhotometricInterpretation, 414
- SetPixelFormat, 414
- SetPlanarConfiguration, 414
- gdcmm::ImageConverter, 415
 - ~ImageConverter, 416
 - Convert, 416
 - GetOutput, 416
 - ImageConverter, 416
 - SetInput, 416
- gdcmm::ImageFragmentSplitter, 416
 - ~ImageFragmentSplitter, 418
 - GetFragmentSizeMax, 418
 - ImageFragmentSplitter, 418
 - SetForce, 418
 - SetFragmentSizeMax, 418
 - Split, 418
- gdcmm::ImageHelper, 418
 - ComputeSpacingFromImagePositionPatient, 419
 - GetDimensionsValue, 419
 - GetDirectionCosinesFromDataSet, 419
 - GetDirectionCosinesValue, 420
 - GetForcePixelSpacing, 420
 - GetForceRescaleInterceptSlope, 420
 - GetLUT, 420
 - GetOriginValue, 420
 - GetPhotometricInterpretationValue, 420
 - GetPixelFormatValue, 420
 - GetPlanarConfigurationValue, 420
 - GetPointerFromElement, 420
 - GetRescaleInterceptSlopeValue, 420
 - GetSpacingTagFromMediaStorage, 420
 - GetSpacingValue, 420
 - GetZSpacingTagFromMediaStorage, 420
 - SetDimensionsValue, 421
 - SetDirectionCosinesValue, 421
 - SetForcePixelSpacing, 421
 - SetForceRescaleInterceptSlope, 421
 - SetOriginValue, 421
 - SetRescaleInterceptSlopeValue, 421
 - SetSpacingValue, 421
- gdcmm::ImageReader, 421
 - ~ImageReader, 424
 - GetImage, 424
 - ImageReader, 424
 - Read, 424
 - ReadACRNEMAImage, 425
 - ReadImage, 425
- gdcmm::ImageToImageFilter, 425
 - ~ImageToImageFilter, 427
 - GetInput, 427
 - GetOutput, 427
 - ImageToImageFilter, 426
- gdcmm::ImageWriter, 427
 - ~ImageWriter, 429
 - GetImage, 429
 - ImageWriter, 429
 - Write, 429
- gdcmm::ImplicitDataElement, 432
 - GetLength, 433
 - Read, 433
 - ReadPreValue, 433
 - ReadValue, 433
 - ReadWithLength, 433
 - Write, 433
- gdcmm::InitializeEvent, 434
- gdcmm::Item, 443
 - Clear, 445
 - FindDataElement, 445
 - GetDataElement, 445
 - GetLength, 445
 - GetNestedDataSet, 445, 446
 - InsertDataElement, 446
 - Item, 445
 - operator<<, 446
 - Read, 446
 - SetNestedDataSet, 446

- Write, 446
- gdcmm::IterationEvent, 446
- gdcmm::JPEG12Codec, 447
 - ~JPEG12Codec, 449
 - Decode, 449
 - GetHeaderInfo, 449
 - InternalCode, 449
 - JPEG12Codec, 449
- gdcmm::JPEG16Codec, 449
 - ~JPEG16Codec, 451
 - Decode, 451
 - GetHeaderInfo, 451
 - InternalCode, 451
 - JPEG16Codec, 451
- gdcmm::JPEG2000Codec, 451
 - ~JPEG2000Codec, 453
 - Bitmap, 454
 - CanCode, 453
 - CanDecode, 453
 - Code, 453
 - Decode, 454
 - GetHeaderInfo, 454
 - GetQuality, 454
 - GetRate, 454
 - JPEG2000Codec, 453
 - SetNumberOfResolutions, 454
 - SetQuality, 454
 - SetRate, 454
 - SetReversible, 454
 - SetTileSize, 454
- gdcmm::JPEG8Codec, 454
 - ~JPEG8Codec, 456
 - Decode, 456
 - GetHeaderInfo, 456
 - InternalCode, 456
 - JPEG8Codec, 456
- gdcmm::JPEGCodec, 456
 - ~JPEGCodec, 458
 - BitSample, 460
 - CanCode, 459
 - CanDecode, 459
 - Code, 459
 - ComputeOffsetTable, 459
 - Decode, 459
 - GetHeaderInfo, 459
 - GetLossless, 459
 - GetQuality, 459
 - IsValid, 459
 - JPEGCodec, 458
 - Lossless, 460
 - Quality, 460
 - SetBitSample, 460
 - SetLossless, 460
 - SetPixelFormat, 460
 - SetQuality, 460
- gdcmm::JPEGLSCodec, 460
 - ~JPEGLSCodec, 462
 - CanCode, 462
 - CanDecode, 462
 - Code, 462
 - Decode, 462, 463
 - GetBufferLength, 463
 - GetHeaderInfo, 463
 - GetLossless, 463
 - JPEGLSCodec, 462
 - SetBufferLength, 463
 - SetLossless, 463
 - SetLossyError, 463
- gdcmm::KAKADUCodec, 463
 - ~KAKADUCodec, 465
 - CanCode, 465
 - CanDecode, 465
 - Code, 465
 - Decode, 465
 - KAKADUCodec, 465
- gdcmm::LO, 465
 - const_iterator, 466
 - const_reference, 466
 - const_reverse_iterator, 466
 - difference_type, 466
 - IsValid, 467
 - iterator, 466
 - LO, 467
 - pointer, 466
 - reference, 466
 - reverse_iterator, 466
 - size_type, 466
 - Superclass, 466
 - value_type, 466
- gdcmm::LookupTable, 467
 - ~LookupTable, 469
 - Allocate, 469
 - BitSample, 471
 - Clear, 470
 - Decode, 470
 - GetBitSample, 470
 - GetBufferAsRGBA, 470
 - GetLUT, 470
 - GetLUTDescriptor, 470
 - GetLUTLength, 470
 - GetPointer, 470
 - IncompleteLUT, 471
 - InitializeBlueLUT, 470
 - InitializeGreenLUT, 470
 - InitializeLUT, 470
 - InitializeRedLUT, 470
 - Initialized, 470
 - Internal, 471

- LookupTable, 469
- LookupTableType, 469
- Print, 470
- SetBlueLUT, 471
- SetGreenLUT, 471
- SetLUT, 471
- SetRedLUT, 471
- WriteBufferAsRGBA, 471
- gdcmm::MD5, 475
 - ~MD5, 476
 - Compute, 476
 - ComputeFile, 476
 - MD5, 476
- gdcmm::Macro, 472
 - AddMacroEntry, 473
 - ArrayIncludeMacrosType, 472
 - Clear, 473
 - FindMacroEntry, 473
 - GetMacroEntry, 473
 - GetName, 473
 - Macro, 472
 - MapModuleEntry, 472
 - operator<<, 473
 - SetName, 473
 - Verify, 473
- gdcmm::Macros, 473
 - AddMacro, 474
 - Clear, 474
 - GetMacro, 474
 - IsEmpty, 474
 - Macros, 474
 - ModuleMapType, 474
 - operator<<, 474
- gdcmm::MediaStorage, 476
 - GetMSString, 481
 - GetMSType, 481
 - GetModality, 481
 - GetModalityDimension, 481
 - GetNumberOfMSString, 482
 - GetNumberOfMSType, 482
 - GetNumberOfModality, 482
 - GetString, 482
 - GuessFromModality, 482
 - IsImage, 482
 - IsUndefined, 482
 - MSType, 479
 - MediaStorage, 481
 - ObjectType, 481
 - operator MSType, 482
 - operator<<, 483
 - SetFromDataSet, 482
 - SetFromFile, 482
 - SetFromHeader, 483
 - SetFromModality, 483
 - SetFromSourceImageSequence, 483
- gdcmm::MemberCommand
 - ~MemberCommand, 485
 - Execute, 485
 - m_ConstMemberFunction, 486
 - m_MemberFunction, 486
 - m_This, 486
 - MemberCommand, 485
 - New, 486
 - Self, 485
 - SetCallbackFunction, 486
 - TConstMemberFunctionPointer, 485
 - TMemberFunctionPointer, 485
- gdcmm::MemberCommand< T >, 483
- gdcmm::MeshPrimitive, 487
 - ~MeshPrimitive, 489
 - AddPrimitiveData, 489
 - GetMPTType, 489
 - GetMPTTypeString, 489
 - GetNumberOfPrimitivesData, 489
 - GetPrimitiveData, 489
 - GetPrimitiveType, 490
 - GetPrimitivesData, 489
 - MPTType, 489
 - MeshPrimitive, 489
 - PrimitiveData, 490
 - PrimitiveType, 490
 - PrimitivesData, 489
 - SetPrimitiveData, 490
 - SetPrimitiveType, 490
 - SetPrimitivesData, 490
- gdcmm::ModifiedEvent, 490
- gdcmm::Module, 491
 - AddMacro, 492
 - AddModuleEntry, 492
 - ArrayIncludeMacrosType, 492
 - Clear, 492
 - FindModuleEntryInMacros, 492
 - GetModuleEntryInMacros, 492
 - GetName, 492
 - MapModuleEntry, 492
 - Module, 492
 - operator<<, 493
 - SetName, 493
 - Verify, 493
- gdcmm::ModuleEntry, 493
 - ~ModuleEntry, 495
 - DataElementType, 496
 - Description, 495
 - DescriptionField, 496
 - GetDescription, 495
 - GetName, 495
 - GetType, 495
 - ModuleEntry, 495

- Name, 496
- operator<<, 495
- SetDescription, 495
- SetName, 495
- SetType, 495
- gdcmm::Modules, 496
 - AddModule, 497
 - Clear, 497
 - GetModule, 497
 - IsEmpty, 497
 - ModuleMapType, 497
 - Modules, 497
 - operator<<, 497
- gdcmm::MovePatientRootQuery, 497
 - GetAbstractSyntaxUID, 499
 - GetTagListByLevel, 499
 - InitializeDataSet, 499
 - MovePatientRootQuery, 499
 - QueryFactory, 499
 - ValidateQuery, 499
- gdcmm::MoveStudyRootQuery, 500
 - GetAbstractSyntaxUID, 501
 - GetTagListByLevel, 501
 - InitializeDataSet, 501
 - MoveStudyRootQuery, 501
 - QueryFactory, 502
 - ValidateQuery, 501
- gdcmm::NestedModuleEntries, 502
 - AddModuleEntry, 504
 - GetModuleEntry, 504
 - GetNumberOfModuleEntries, 504
 - NestedModuleEntries, 504
 - operator<<, 504
 - SizeType, 504
- gdcmm::NoEvent, 504
- gdcmm::Object, 505
 - ~Object, 507
 - Object, 507
 - operator<<, 507
 - operator=, 507
 - Print, 507
 - Register, 507
 - SmartPointer, 507
 - UnRegister, 507
- gdcmm::OneShotReadBuf, 508
 - OneShotReadBuf, 508
- gdcmm::Orientation, 508
 - ~Orientation, 509
 - GetLabel, 509
 - GetMajorAxisFromPatientRelativeDirectionCosine, 509
 - GetObliquityThresholdCosineValue, 509
 - GetType, 509
 - operator<<, 510
 - Orientation, 509
 - OrientationType, 509
 - Print, 509
 - SetObliquityThresholdCosineValue, 510
- gdcmm::Overlay, 510
 - ~Overlay, 513
 - Decode, 513
 - Decompress, 513
 - GetBitPosition, 513
 - GetBitsAllocated, 513
 - GetBuffer, 513
 - GetColumns, 513
 - GetDescription, 513
 - GetGroup, 513
 - GetOrigin, 513
 - GetOverlayData, 513
 - GetRows, 513
 - GetType, 513
 - GetUnpackBuffer, 514
 - GrabOverlayFromPixelData, 514
 - IsEmpty, 514
 - IsInPixelData, 514
 - IsZero, 514
 - Overlay, 513
 - Print, 514
 - SetBitPosition, 514
 - SetBitsAllocated, 514
 - SetColumns, 514
 - SetDescription, 514
 - SetFrameOrigin, 514
 - SetGroup, 514
 - SetNumberOfFrames, 514
 - SetOrigin, 515
 - SetOverlay, 515
 - SetRows, 515
 - SetType, 515
 - Update, 515
- gdcmm::PDBelement, 522
 - GetName, 523
 - GetValue, 523
 - NameField, 524
 - operator<<, 524
 - operator==, 523
 - PDBelement, 523
 - SetName, 523
 - SetValue, 524
 - ValueField, 524
- gdcmm::PDBHeader, 524
 - ~PDBHeader, 525
 - FindPDBelementByName, 525
 - GetPDBelementEnd, 525
 - GetPDBelementByName, 525
 - GetPDBInfoTag, 525
 - LoadFromDataElement, 525

- operator<<, 526
- PDBHeader, 525
- Print, 525
- gdcmm::PDFCodec, 526
 - ~PDFCodec, 527
 - CanCode, 527
 - CanDecode, 527
 - Decode, 528
 - PDFCodec, 527
- gdcmm::PNMCodec, 549
 - ~PNMCodec, 551
 - CanCode, 551
 - CanDecode, 551
 - GetBufferLength, 551
 - GetHeaderInfo, 551
 - PNMCodec, 551
 - Read, 551
 - SetBufferLength, 551
 - Write, 551
- gdcmm::PVRGCodec, 571
 - ~PVRGCodec, 572
 - CanCode, 572
 - CanDecode, 572
 - Code, 572
 - Decode, 573
 - PVRGCodec, 572
- gdcmm::ParseException, 515
 - ~ParseException, 517
 - GetLastElement, 517
 - operator=, 517
 - ParseException, 517
 - SetLastElement, 517
- gdcmm::Parser, 519
 - ~Parser, 519
 - EndElementHandler, 518
 - ErrorType, 518
 - GetBuffer, 519
 - GetCurrentByteIndex, 519
 - GetErrorCode, 519
 - GetErrorString, 519
 - GetUserData, 519
 - Parse, 519
 - ParseBuffer, 519
 - Parser, 519
 - Process, 519
 - SetElementHandler, 519
 - SetUserData, 519
 - StartElementHandler, 518
- gdcmm::Patient, 519
 - Patient, 520
- gdcmm::PersonName, 529
 - Component, 530
 - GetMaxLength, 530
 - GetNumberOfComponents, 530
 - MaxLength, 530
 - MaxNumberOfComponents, 530
 - Padding, 530
 - Print, 530
 - Separator, 530
 - SetBlob, 530
 - SetComponents, 530
- gdcmm::PhotometricInterpretation, 530
 - GetPIString, 532
 - GetPIType, 532
 - GetSamplesPerPixel, 532
 - GetString, 532
 - GetType, 532
 - IsLossless, 532
 - IsLossy, 532
 - IsRetired, 532
 - IsSameColorSpace, 532
 - operator PType, 533
 - operator<<, 533
 - PType, 532
 - PhotometricInterpretation, 532
- gdcmm::PixelFormat, 533
 - ~PixelFormat, 535
 - Bitmap, 537
 - GetBitsAllocated, 535
 - GetBitsStored, 535
 - GetHighBit, 535
 - GetMax, 536
 - GetMin, 536
 - GetPixelRepresentation, 536
 - GetPixelSize, 536
 - GetSamplesPerPixel, 536
 - GetScalarType, 536
 - GetScalarTypeAsString, 536
 - IsValid, 536
 - operator ScalarType, 536
 - operator<<, 537
 - operator==, 537
 - PixelFormat, 535
 - Print, 537
 - ScalarType, 535
 - SetBitsAllocated, 537
 - SetBitsStored, 537
 - SetHighBit, 537
 - SetPixelRepresentation, 537
 - SetSamplesPerPixel, 537
 - SetScalarType, 537
 - Validate, 537
- gdcmm::Pixmap, 538
 - ~Pixmap, 540
 - AreOverlaysInImageData, 540
 - Curves, 541
 - GetCurve, 540
 - GetIconImage, 540

- GetNumberOfCurves, 540
- GetNumberOfOverlays, 540
- GetOverlay, 540
- Icon, 541
- Overlays, 541
- Pixmap, 540
- Print, 540
- RemoveOverlay, 541
- SetIconImage, 541
- SetNumberOfCurves, 541
- SetNumberOfOverlays, 541
- gdcm::PixmapReader, 541
 - ~PixmapReader, 543
 - GetPixmap, 543
 - ImageData, 544
 - PixmapReader, 543
 - Read, 543
 - ReadACRNEMAImage, 543
 - ReadImage, 543
- gdcm::PixmapToPixmapFilter, 544
 - ~PixmapToPixmapFilter, 545
 - GetInput, 546
 - GetOutput, 546
 - PixmapToPixmapFilter, 545
- gdcm::PixmapWriter, 546
 - ~PixmapWriter, 548
 - DolconImage, 548
 - GetImage, 548
 - GetPixmap, 548, 549
 - ImageData, 549
 - PixmapWriter, 548
 - PrepareWrite, 549
 - SetImage, 549
 - SetPixmap, 549
 - Write, 549
- gdcm::Preamble, 552
 - ~Preamble, 552
 - Clear, 553
 - Create, 553
 - GetInternal, 553
 - GetLength, 553
 - IsEmpty, 553
 - IsValid, 553
 - operator<<, 553
 - operator=, 553
 - Preamble, 552
 - Print, 553
 - Read, 553
 - Remove, 553
 - Valid, 553
 - Write, 553
- gdcm::PresentationContext, 553
 - AddTransferSyntax, 554
 - GetAbstractSyntax, 554
 - GetNumberOfTransferSyntaxes, 554
 - GetPresentationContextID, 554
 - GetTransferSyntax, 554
 - operator==, 554
 - PresentationContext, 554
 - Print, 554
 - SetAbstractSyntax, 555
 - SetPresentationContextID, 555
 - SizeType, 554
 - TransferSyntaxArrayType, 554
- gdcm::PresentationContextGenerator, 556
 - AddPresentationContext, 557
 - GenerateFromFilenames, 557
 - GenerateFromUID, 557
 - GetDefaultTransferSyntax, 557
 - GetPresentationContexts, 558
 - PresentationContextArrayType, 557
 - PresentationContextGenerator, 557
 - SetDefaultTransferSyntax, 558
 - SetMergeModeToAbstractSyntax, 558
 - SetMergeModeToTransferSyntax, 558
 - SizeType, 557
- gdcm::Printer, 562
 - ~Printer, 564
 - F, 565
 - GetPrintStyle, 564
 - MaxPrintLength, 565
 - Print, 564
 - PrintDataElement, 564
 - PrintDataSet, 564
 - PrintSQ, 564
 - PrintStyle, 565
 - PrintStyles, 564
 - Printer, 564
 - SetColor, 565
 - SetFile, 565
 - SetStyle, 565
- gdcm::PrivateDict, 565
 - ~PrivateDict, 566
 - AddDictEntry, 566
 - Dicts, 566
 - FindDictEntry, 566
 - GetDictEntry, 566
 - IsEmpty, 566
 - LoadDefault, 566
 - operator<<, 566
 - PrintXML, 566
 - PrivateDict, 566
 - RemoveDictEntry, 566
- gdcm::PrivateTag, 567
 - GetOwner, 568
 - operator<, 568
 - operator<<, 568
 - PrivateTag, 568

- ReadFromCommaSeparatedString, 568
- SetOwner, 568
- gdcmm::ProgressEvent, 569
 - ~ProgressEvent, 570
 - CheckEvent, 570
 - GetEventName, 570
 - GetProgress, 570
 - MakeObject, 570
 - ProgressEvent, 570
 - Self, 570
 - SetProgress, 570
 - Superclass, 570
- gdcmm::PythonFilter, 573
 - ~PythonFilter, 573
 - GetFile, 573
 - PythonFilter, 573
 - SetDicts, 574
 - SetFile, 574
 - ToPyObject, 574
 - UseDictAlways, 574
- gdcmm::QueryBase, 574
 - ~QueryBase, 575
 - GetAllTags, 575
 - GetName, 575
 - GetOptionalTags, 575
 - GetQueryLevel, 575
 - GetRequiredTags, 575
 - GetUniqueTags, 575
- gdcmm::QueryFactory, 576
 - GetCharacterFromCurrentLocale, 576
 - ListCharSets, 576
 - ProduceCharacterSetDataElement, 576
 - ProduceQuery, 576
- gdcmm::QueryImage, 577
 - GetName, 578
 - GetOptionalTags, 578
 - GetQueryLevel, 578
 - GetRequiredTags, 578
 - GetUniqueTags, 578
- gdcmm::QueryPatient, 578
 - GetName, 579
 - GetOptionalTags, 579
 - GetQueryLevel, 580
 - GetRequiredTags, 580
 - GetUniqueTags, 580
- gdcmm::QuerySeries, 580
 - GetName, 581
 - GetOptionalTags, 581
 - GetQueryLevel, 581
 - GetRequiredTags, 581
 - GetUniqueTags, 581
- gdcmm::QueryStudy, 582
 - GetName, 583
 - GetOptionalTags, 583
- GetQueryLevel, 583
- GetRequiredTags, 583
- GetUniqueTags, 583
- gdcmm::RAWCodec, 583
 - ~RAWCodec, 585
 - CanCode, 585
 - CanDecode, 585
 - Code, 585
 - Decode, 585
 - DecodeBytes, 585
 - GetHeaderInfo, 586
 - RAWCodec, 585
- gdcmm::RLECodec, 594
 - ~RLECodec, 595
 - CanCode, 595
 - CanDecode, 595
 - Code, 596
 - Decode, 596
 - GetBufferLength, 596
 - GetHeaderInfo, 596
 - RLECodec, 595
 - SetBufferLength, 596
 - SetLength, 596
- gdcmm::Reader, 586
 - ~Reader, 589
 - CanRead, 589
 - F, 591
 - GetFile, 589
 - GetStreamPtr, 589
 - Read, 589
 - ReadDataSet, 589
 - ReadMetaInformation, 590
 - ReadPreamble, 590
 - ReadSelectedTags, 590
 - ReadUpToTag, 590
 - Reader, 589
 - SetFile, 590
 - SetFileName, 590
 - SetStream, 590
 - StreamImageReader, 590
- gdcmm::Rescaler, 591
 - ~Rescaler, 592
 - ComputeInterceptSlopePixelType, 592
 - ComputePixelTypeFromMinMax, 593
 - GetIntercept, 593
 - GetSlope, 593
 - InverseRescale, 593
 - InverseRescaleFunctionIntoBestFit, 593
 - Rescale, 593
 - RescaleFunctionIntoBestFit, 593
 - Rescaler, 592
 - SetIntercept, 593
 - SetMinMaxForPixelType, 593
 - SetPixelFormat, 593

- SetSlope, 593
- SetTargetPixelType, 593
- SetUseTargetPixelType, 593
- gdcm::SHA1, 633
 - ~SHA1, 633
 - Compute, 633
 - ComputeFile, 634
 - SHA1, 633
- gdcm::SOPClassUIDToIOD, 642
 - const, 642
 - GetIOD, 642
- gdcm::Scanner, 597
 - ~Scanner, 601
 - AddPrivateTag, 601
 - AddSkipTag, 601
 - AddTag, 601
 - Begin, 601
 - ClearSkipTags, 601
 - ClearTags, 601
 - ConstIterator, 600
 - End, 601
 - GetAllFileNamesFromTagToValue, 601
 - GetFilenameFromTagToValue, 601
 - GetFileNames, 601
 - GetKeys, 601
 - GetMapping, 602
 - GetMappingFromTagToValue, 602
 - GetMappings, 602
 - GetOrderedValues, 602
 - GetValue, 602
 - GetValues, 602
 - IsKey, 602
 - MappingType, 600
 - New, 603
 - operator<<, 603
 - Print, 603
 - ProcessPublicTag, 603
 - Scan, 603
 - Scanner, 601
 - TagToValue, 600
 - TagToValueValueType, 600
 - ValueType, 601
- gdcm::Scanner::ltstr, 471
 - operator(), 471
- gdcm::Segment, 603
 - ~Segment, 606
 - ALGOType, 606
 - AddSurface, 606
 - AnatomicRegion, 607
 - GetALGOType, 606
 - GetALGOTypeString, 606
 - GetAnatomicRegion, 606
 - GetPropertyCategory, 606
 - GetPropertyType, 606
 - GetSegmentAlgorithmName, 606
 - GetSegmentAlgorithmType, 606
 - GetSegmentDescription, 606
 - GetSegmentLabel, 606
 - GetSegmentNumber, 606
 - GetSurface, 606
 - GetSurfaceCount, 606
 - GetSurfaces, 607
 - PropertyCategory, 607
 - PropertyType, 607
 - Segment, 606
 - SegmentAlgorithmName, 607
 - SegmentAlgorithmType, 607
 - SegmentDescription, 607
 - SegmentLabel, 607
 - SegmentNumber, 607
 - SetAnatomicRegion, 607
 - SetPropertyCategory, 607
 - SetPropertyType, 607
 - SetSegmentAlgorithmName, 607
 - SetSegmentAlgorithmType, 607
 - SetSegmentDescription, 607
 - SetSegmentLabel, 607
 - SetSegmentNumber, 607
 - SetSurfaceCount, 607
 - SurfaceCount, 607
 - SurfaceVector, 606
 - Surfaces, 607
- gdcm::SegmentHelper, 129
- gdcm::SegmentHelper::BasicCodedEntry, 196
 - BasicCodedEntry, 198
 - CM, 198
 - CSD, 198
 - CSV, 198
 - CV, 198
 - IsEmpty, 198
- gdcm::SegmentReader, 609
 - ~SegmentReader, 611
 - GetSegments, 611
 - Read, 612
 - ReadSegment, 612
 - ReadSegments, 612
 - SegmentMap, 611
 - SegmentReader, 611
 - SegmentVector, 611
 - Segments, 612
- gdcm::SegmentWriter, 612
 - ~SegmentWriter, 614
 - AddSegment, 614
 - GetNumberOfSegments, 614
 - GetSegment, 614
 - GetSegments, 614
 - PrepareWrite, 614
 - SegmentVector, 614

- SegmentWriter, 614
- Segments, 614
- SetNumberOfSegments, 614
- SetSegments, 614
- Write, 614
- gdcmm::SegmentedPaletteColorLookupTable, 608
 - ~SegmentedPaletteColorLookupTable, 609
 - Print, 609
 - SegmentedPaletteColorLookupTable, 609
 - SetLUT, 609
- gdcmm::SequenceOfFragments, 614
 - AddFragment, 617
 - Begin, 617
 - Clear, 617
 - ComputeByteLength, 617
 - ComputeLength, 617
 - ConstIterator, 617
 - End, 617
 - FragmentVector, 617
 - GetBuffer, 617
 - GetFragBuffer, 617
 - GetFragment, 617
 - GetLength, 618
 - GetNumberOfFragments, 618
 - GetTable, 618
 - Iterator, 617
 - New, 618
 - operator==, 618
 - Print, 618
 - Read, 618
 - SequenceOfFragments, 617
 - SetLength, 618
 - SizeType, 617
 - Write, 618
 - WriteBuffer, 618
- gdcmm::SequenceOfItems, 619
 - AddItem, 622
 - Begin, 622
 - Clear, 622
 - ComputeLength, 622
 - ConstIterator, 622
 - End, 622
 - FindDataElement, 622
 - GetItem, 622
 - GetLength, 623
 - GetNumberOfItems, 623
 - IsUndefinedLength, 623
 - ItemVector, 622
 - Items, 624
 - Iterator, 622
 - New, 623
 - operator=, 623
 - operator==, 623
 - Print, 623
 - Read, 623
 - SequenceLengthField, 624
 - SequenceOfItems, 622
 - SetLength, 623
 - SetLengthToUndefined, 624
 - SetNumberOfItems, 624
 - SizeType, 622
 - Write, 624
- gdcmm::SerieHelper, 624
 - ~SerieHelper, 626
 - AddFile, 626
 - AddFileName, 626
 - AddRestriction, 626
 - Clear, 627
 - CreateDefaultUniqueSeriesIdentifier, 627
 - CreateUniqueSeriesIdentifier, 627
 - FileNameOrdering, 627
 - GetFirstSingleSerieUIDFileSet, 627
 - GetNextSingleSerieUIDFileSet, 627
 - ImagePositionPatientOrdering, 627
 - ItFileSetHt, 627
 - OrderFileList, 627
 - SerieHelper, 626
 - SerieRestrictions, 626
 - SetDirectory, 627
 - SetLoadMode, 627
 - SetUseSeriesDetails, 627
 - SingleSerieUIDFileSetHT, 627
 - SingleSerieUIDFileSetmap, 626
 - UserOrdering, 627
- gdcmm::SerieHelper::Rule, 596
 - elem, 597
 - group, 597
 - op, 597
 - value, 597
- gdcmm::Series, 627
 - Series, 628
- gdcmm::ServiceClassUser, 628
 - ~ServiceClassUser, 630
 - GetAETitle, 630
 - GetCalledAETitle, 630
 - GetTimeout, 630
 - InitializeConnection, 630
 - SendEcho, 630
 - SendFind, 631
 - SendMove, 631
 - SendStore, 631
 - ServiceClassUser, 630
 - SetAETitle, 631
 - SetCalledAETitle, 631
 - SetHostname, 631
 - SetPort, 632
 - SetPortSCP, 632
 - SetPresentationContexts, 632

- SetTimeout, 632
- StartAssociation, 632
- StopAssociation, 632
- gdcmm::SimpleMemberCommand
 - ~SimpleMemberCommand, 636
 - Execute, 636
 - m_MemberFunction, 637
 - m_This, 637
 - New, 637
 - Self, 636
 - SetCallbackFunction, 637
 - SimpleMemberCommand, 636
 - TMemberFunctionPointer, 636
- gdcmm::SimpleMemberCommand< T >, 634
- gdcmm::SimpleSubjectWatcher, 637
 - ~SimpleSubjectWatcher, 638
 - EndFilter, 638
 - ShowAbort, 638
 - ShowAnonymization, 638
 - ShowData, 638
 - ShowDataSet, 638
 - ShowIteration, 638
 - ShowProgress, 638
 - SimpleSubjectWatcher, 638
 - StartFilter, 638
 - TestAbortOff, 638
 - TestAbortOn, 638
- gdcmm::SmartPointer
 - ~SmartPointer, 641
 - GetPointer, 641
 - operator ObjectType *, 641
 - operator*, 641
 - operator->, 641
 - operator=, 641
 - SmartPointer, 641
- gdcmm::SmartPointer< ObjectType >, 639
- gdcmm::Sorter, 643
 - ~Sorter, 645
 - AddSelect, 645
 - FileNames, 646
 - GetFileNames, 645
 - operator<<, 646
 - Print, 646
 - Selection, 646
 - SelectionMap, 645
 - SetSortFunction, 646
 - Sort, 646
 - SortFunc, 646
 - SortFunction, 645
 - Sorter, 645
 - StableSort, 646
- gdcmm::Spacing, 647
 - ~Spacing, 648
 - ComputePixelAspectRatioFromPixelSpacing, 648
 - Spacing, 648
 - SpacingType, 648
- gdcmm::Spectroscopy, 649
 - Spectroscopy, 649
- gdcmm::SplitMosaicFilter, 649
 - ~SplitMosaicFilter, 650
 - ComputeMOSAICDimensions, 650
 - GetFile, 650
 - GetImage, 650
 - SetFile, 650
 - SetImage, 650
 - Split, 650
 - SplitMosaicFilter, 650
- gdcmm::StartEvent, 650
- gdcmm::StreamImageReader, 652
 - ~StreamImageReader, 654
 - CanReadImage, 654
 - DefinePixelExtent, 654
 - DefineProperBufferLength, 655
 - GetDimensionsValueForResolution, 655
 - GetFile, 655
 - mFileOffset, 656
 - mFileOffset1, 656
 - mHeaderInformation, 656
 - mReader, 656
 - mXMax, 656
 - mXMin, 656
 - mYMax, 656
 - mYMin, 656
 - mZMax, 656
 - mZMin, 657
 - Read, 655
 - ReadImageInformation, 655
 - ReadImageSubregionJpegLS, 656
 - ReadImageSubregionRAW, 656
 - SetFileName, 656
 - SetStream, 656
 - StreamImageReader, 654
- gdcmm::StreamImageWriter, 657
 - ~StreamImageWriter, 659
 - CanWriteFile, 659
 - DefinePixelExtent, 660
 - DefineProperBufferLength, 660
 - mElementOffsets, 661
 - mElementOffsets1, 661
 - mWriter, 662
 - mXMax, 662
 - mXMin, 662
 - mYMax, 662
 - mYMin, 662
 - mZMax, 662
 - mZMin, 662
 - mspFile, 662
 - SetFile, 660

- SetFileName, 660
- SetStream, 660
- StreamImageWriter, 659
- Write, 660
- WriteImageInformation, 661
- WriteImageSubregionRAW, 661
- WriteRawHeader, 661
- gdcmm::String
 - const_iterator, 664
 - const_reference, 664
 - const_reverse_iterator, 664
 - difference_type, 664
 - IsValid, 665
 - iterator, 664
 - operator const char *, 665
 - pointer, 664
 - reference, 664
 - reverse_iterator, 664
 - size_type, 664
 - String, 664, 665
 - Trim, 665
 - Truncate, 665
 - value_type, 664
- gdcmm::String< TDelimiter, TMaxLength, TPadChar >, 662
- gdcmm::StringFilter, 665
 - ~StringFilter, 666
 - ExecuteQuery, 666
 - FromString, 667
 - GetFile, 667
 - SetDicts, 667
 - SetFile, 667
 - StringFilter, 666
 - ToString, 667
 - ToStringPair, 667
 - UseDictAlways, 667
- gdcmm::Study, 668
 - Study, 668
- gdcmm::Subject, 668
 - ~Subject, 669
 - AddObserver, 669, 670
 - GetCommand, 670
 - HasObserver, 670
 - InvokeEvent, 670
 - RemoveAllObservers, 670
 - RemoveObserver, 670
 - Subject, 669
- gdcmm::Surface, 670
 - ~Surface, 674
 - GetAlgorithmFamily, 674
 - GetAlgorithmName, 674
 - GetAlgorithmVersion, 674
 - GetAxisOfRotation, 674
 - GetCenterOfRotation, 674
 - GetFiniteVolume, 674
 - GetManifold, 674
 - GetMaximumPointDistance, 674
 - GetMeanPointDistance, 674
 - GetMeshPrimitive, 674, 675
 - GetNumberOfSurfacePoints, 675
 - GetNumberOfVectors, 675
 - GetPointCoordinatesData, 675
 - GetPointPositionAccuracy, 675
 - GetPointsBoundingBoxCoordinates, 675
 - GetProcessingAlgorithm, 675
 - GetRecommendedDisplayCIELabValue, 675
 - GetRecommendedDisplayGrayscaleValue, 675
 - GetRecommendedPresentationOpacity, 675
 - GetRecommendedPresentationType, 675
 - GetSTATES, 675
 - GetSTATESString, 675
 - GetSurfaceComments, 675
 - GetSurfaceNumber, 675
 - GetSurfaceProcessing, 675
 - GetSurfaceProcessingDescription, 675
 - GetSurfaceProcessingRatio, 675
 - GetVIEWType, 676
 - GetVIEWTypeString, 676
 - GetVectorAccuracy, 675
 - GetVectorCoordinateData, 676
 - GetVectorDimensionality, 676
 - STATES, 673
 - SetAlgorithmFamily, 676
 - SetAlgorithmName, 676
 - SetAlgorithmVersion, 676
 - SetAxisOfRotation, 676
 - SetCenterOfRotation, 676
 - SetFiniteVolume, 676
 - SetManifold, 676
 - SetMaximumPointDistance, 676
 - SetMeanPointDistance, 676
 - SetMeshPrimitive, 676
 - SetNumberOfSurfacePoints, 676
 - SetNumberOfVectors, 676
 - SetPointCoordinatesData, 676
 - SetPointPositionAccuracy, 676
 - SetPointsBoundingBoxCoordinates, 676
 - SetProcessingAlgorithm, 676
 - SetRecommendedDisplayCIELabValue, 676
 - SetRecommendedDisplayGrayscaleValue, 676
 - SetRecommendedPresentationOpacity, 676
 - SetRecommendedPresentationType, 677
 - SetSurfaceComments, 677
 - SetSurfaceNumber, 677
 - SetSurfaceProcessing, 677
 - SetSurfaceProcessingDescription, 677
 - SetSurfaceProcessingRatio, 677
 - SetVectorAccuracy, 677
 - SetVectorCoordinateData, 677

- SetVectorDimensionality, 677
- Surface, 674
- VIEWType, 673
- gdcmm::SurfaceHelper, 677
 - ColorArray, 678
 - RGBToRecommendedDisplayCIELab, 678
 - RGBToRecommendedDisplayGrayscale, 679
 - RecommendedDisplayCIELabToRGB, 678
- gdcmm::SurfaceReader, 679
 - ~SurfaceReader, 681
 - GetNumberOfSurfaces, 681
 - Read, 681
 - ReadPointMacro, 681
 - ReadSurface, 681
 - ReadSurfaces, 681
 - SurfaceReader, 681
- gdcmm::SurfaceWriter, 682
 - ~SurfaceWriter, 683
 - ComputeNumberOfSurfaces, 683
 - GetNumberOfSurfaces, 683
 - NumberOfSurfaces, 683
 - PrepareWrite, 683
 - PrepareWritePointMacro, 683
 - SetNumberOfSurfaces, 683
 - SurfaceWriter, 683
 - Write, 683
- gdcmm::SwapCode, 684
 - GetIndex, 685
 - GetSwapCodeString, 685
 - operator SwapCode::SwapCodeType, 685
 - operator<<, 685
 - SwapCode, 685
 - SwapCodeType, 684
- gdcmm::SwapperDoOp, 685
 - Swap, 685
 - SwapArray, 685
- gdcmm::SwapperNoOp, 686
 - Swap, 686
 - SwapArray, 686
- gdcmm::System, 686
 - DeleteDirectory, 687
 - EncodeBytes, 687
 - FileExists, 687
 - FileIsDirectory, 688
 - FileIsSymlink, 688
 - FileSize, 688
 - FileTime, 688
 - FormatDateTime, 688
 - GetCWD, 689
 - GetCurrentDateTime, 688
 - GetCurrentModuleFileName, 688
 - GetCurrentProcessFileName, 689
 - GetCurrentResourcesDirectory, 689
 - GetHostName, 689
 - GetLastSystemError, 689
 - GetLocaleCharset, 689
 - GetPermissions, 689
 - GetTimezoneOffsetFromUTC, 689
 - MakeDirectory, 689
 - ParseDateTime, 689, 690
 - RemoveFile, 690
 - SetPermissions, 690
 - StrCaseCmp, 690
 - StrNCaseCmp, 690
 - StrTokR, 690
- gdcmm::Table, 690
 - ~Table, 691
 - GetTableEntry, 691
 - InsertEntry, 691
 - MapTableEntry, 691
 - operator<<, 691
 - Table, 691
- gdcmm::TableEntry, 691
 - ~TableEntry, 692
 - TableEntry, 692
- gdcmm::TableReader, 692
 - ~TableReader, 693
 - CharacterDataHandler, 693
 - EndElement, 693
 - GetDefs, 693
 - GetFilename, 693
 - HandleIOD, 693
 - HandleIODEntry, 693
 - HandleMacro, 693
 - HandleMacroEntry, 693
 - HandleMacroEntryDescription, 693
 - HandleModule, 693
 - HandleModuleEntry, 693
 - HandleModuleEntryDescription, 693
 - HandleModuleInclude, 694
 - Read, 694
 - SetFilename, 694
 - StartElement, 694
 - TableReader, 693
- gdcmm::Tag, 695
 - bytes, 701
 - GetElement, 697
 - GetElementTag, 697
 - GetGroup, 697
 - GetLength, 698
 - GetPrivateCreator, 698
 - IsGroupLength, 698
 - IsGroupXX, 698
 - IsIllegal, 698
 - IsPrivate, 698
 - IsPrivateCreator, 698
 - IsPublic, 699
 - operator<, 699

- operator<<, 701
- operator<=, 699
- operator>>, 701
- operator=, 699
- operator==, 699
- PrintAsPipeSeparatedString, 699
- Read, 700
- ReadFromCommaSeparatedString, 700
- ReadFromPipeSeparatedString, 700
- SetElement, 700
- SetElementTag, 700
- SetGroup, 700
- SetPrivateCreator, 700
- Tag, 697
- tag, 701
- tags, 701
- Write, 701
- gdcmm::TagPath, 701
 - ~TagPath, 702
 - ConstructFromString, 702
 - ConstructFromTagList, 702
 - IsValid, 702
 - Print, 702
 - Push, 702
 - TagPath, 702
- gdcmm::Testing, 703
 - ~Testing, 704
 - ComputeFileMD5, 704
 - ComputeMD5, 704
 - GetDataExtraRoot, 704
 - GetDataRoot, 704
 - GetFileName, 705
 - GetFileNames, 705
 - GetLossyFlagFromFile, 705
 - GetMD5DataImage, 705
 - GetMD5DataImages, 705
 - GetMD5FromBrokenFile, 705
 - GetMD5FromFile, 705
 - GetMediaStorageDataFile, 705
 - GetMediaStorageDataFiles, 705
 - GetMediaStorageFromFile, 705
 - GetNumberOfFileNames, 705
 - GetNumberOfMD5DataImages, 705
 - GetNumberOfMediaStorageDataFiles, 705
 - GetPixelSpacingDataRoot, 706
 - GetSelectedTagsOffsetFromFile, 706
 - GetSourceDirectory, 706
 - GetStreamOffsetFromFile, 706
 - GetTempDirectory, 706
 - GetTempDirectoryW, 706
 - GetTempFilename, 706
 - GetTempFilenameW, 706
 - MD5DataImagesType, 704
 - MediaStorageDataFilesType, 704
 - Print, 706
 - Testing, 704
- gdcmm::Trace, 706
 - ~Trace, 707
 - DebugOff, 707
 - DebugOn, 707
 - ErrorOff, 707
 - ErrorOn, 708
 - GetDebugFlag, 708
 - GetErrorFlag, 708
 - GetStream, 708
 - GetWarningFlag, 708
 - SetDebug, 708
 - SetError, 708
 - SetStream, 708
 - SetWarning, 708
 - Trace, 707
 - WarningOff, 708
 - WarningOn, 708
- gdcmm::TransferSyntax, 708
 - CanStoreLossy, 711
 - GetNegociatedType, 711
 - GetString, 711
 - GetSwapCode, 711
 - GetTSSString, 711
 - GetTSType, 711
 - IsEncapsulated, 712
 - IsEncoded, 712
 - IsExplicit, 712
 - IsImplicit, 712
 - IsLossless, 712
 - IsLossy, 712
 - IsValid, 712
 - NegociatedType, 710
 - operator TSType, 712
 - operator<<, 712
 - TSType, 710
 - TransferSyntax, 711
- gdcmm::Type, 715
 - GetTypeString, 716
 - GetTypeType, 716
 - operator TypeType, 716
 - operator<<, 717
 - Type, 716
 - TypeType, 716
- gdcmm::UI, 717
 - Internal, 717
 - operator<<, 717
- gdcmm::UIDGenerator, 717
 - Generate, 718
 - GenerateUUID, 718
 - GetGDCMUID, 718
 - GetRoot, 718
 - IsValid, 719

- SetRoot, 719
- UIDGenerator, 718
- gdcmm::UIDs, 719
 - GetName, 737
 - GetNumberOfTransferSyntaxStrings, 737
 - GetString, 738
 - GetTransferSyntaxString, 738
 - GetTransferSyntaxStrings, 738
 - GetUIDName, 738
 - GetUIDString, 738
 - operator TSType, 738
 - SetFromUID, 738
 - TSName, 724
 - TSType, 731
 - TransferSyntaxStringsType, 724
- gdcmm::UNExplicitDataElement, 786
 - GetLength, 787
 - Read, 787
 - ReadPreValue, 787
 - ReadValue, 787
 - ReadWithLength, 787
- gdcmm::UNExplicitImplicitDataElement, 787
 - GetLength, 789
 - Read, 789
 - ReadPreValue, 789
 - ReadValue, 789
- gdcmm::Unpacker12Bits, 789
 - Pack, 790
 - Unpack, 790
- gdcmm::Usage, 790
 - GetUsageString, 791
 - GetUsageType, 791
 - operator UsageType, 792
 - operator<<, 792
 - Usage, 791
 - UsageType, 791
- gdcmm::UserEvent, 792
- gdcmm::VL, 799
 - GetLength, 800
 - GetVL16Max, 800
 - GetVL32Max, 800
 - IsOdd, 800
 - IsUndefined, 800
 - operator uint32_t, 800
 - operator<<, 801
 - operator++, 801
 - operator+==, 801
 - Read, 801
 - Read16, 801
 - SetToUndefined, 801
 - Type, 800
 - VL, 800
 - Write, 801
 - Write16, 801
- gdcmm::VM, 801
 - Compatible, 804
 - GetIndex, 804
 - GetLength, 804
 - GetNumberOfElementsFromArray, 804
 - GetVMString, 804
 - GetVMType, 804
 - GetVMTypeFromLength, 805
 - IsValid, 805
 - operator VMType, 805
 - operator<<, 805
 - VM, 804
 - VMType, 803
- gdcmm::VR, 805
 - CanDisplay, 808
 - Compatible, 808
 - GetLength, 808, 809
 - GetSize, 809
 - GetSizeof, 809
 - GetVRString, 809
 - GetVRStringFromFile, 809
 - GetVRType, 809
 - GetVRTypeFromFile, 809
 - IsASCII, 809
 - IsASCII2, 809
 - IsBinary, 809
 - IsBinary2, 809
 - IsDual, 809
 - IsSwap, 809
 - IsVRFile, 809
 - IsValid, 809
 - operator VRType, 809
 - operator<<, 810
 - Read, 809
 - VR, 808
 - VRType, 807
 - Write, 809
- gdcmm::VR16ExplicitDataElement, 810
 - GetLength, 811
 - Read, 811
 - ReadPreValue, 812
 - ReadValue, 812
 - ReadWithLength, 812
- gdcmm::VRVLSIZE< 0 >, 812
 - Read, 812
 - Write, 812
- gdcmm::VRVLSIZE< 1 >, 812
 - Read, 813
 - Write, 813
- gdcmm::Validate, 794
 - ~Validate, 795
 - F, 795
 - GetValidatedFile, 795
 - SetFile, 795

- V, 795
- Validate, 795
- Validation, 795
- gdcmm::Value, 795
 - ~Value, 797
 - Clear, 797
 - GetLength, 797
 - operator==, 797
 - SetLength, 797
 - Value, 797
- gdcmm::ValueIO
 - Read, 798
 - Write, 798
- gdcmm::ValueIO< TDE, TSwap, TType >, 797
- gdcmm::Version, 798
 - ~Version, 798
 - GetBuildVersion, 798
 - GetMajorVersion, 799
 - GetMinorVersion, 799
 - GetVersion, 799
 - operator<<, 799
 - Print, 799
 - Version, 798
- gdcmm::Waveform, 854
 - Waveform, 854
- gdcmm::Writer, 854
 - ~Writer, 858
 - CheckFileMetaInformationOff, 858
 - CheckFileMetaInformationOn, 858
 - GetFile, 858
 - GetStreamPtr, 858
 - Ofstream, 859
 - SetCheckFileMetaInformation, 858
 - SetFile, 858
 - SetFileName, 859
 - SetStream, 859
 - SetWriteDataSetOnly, 859
 - Stream, 859
 - StreamImageWriter, 859
 - Write, 859
 - Writer, 858
- gdcmm::XMLDictReader, 860
 - ~XMLDictReader, 861
 - CharacterDataHandler, 861
 - EndElement, 861
 - GetDict, 861
 - HandleDescription, 861
 - HandleEntry, 861
 - StartElement, 861
 - XMLDictReader, 861
- gdcmm::XMLPrivateDictReader, 861
 - ~XMLPrivateDictReader, 863
 - CharacterDataHandler, 863
 - EndElement, 863
 - GetPrivateDict, 863
 - HandleDescription, 863
 - HandleEntry, 863
 - StartElement, 863
 - XMLPrivateDictReader, 863
- gdcmm::ignore_char, 392
 - ignore_char, 393
 - m_char, 393
- gdcmm::network, 124
 - cMaxEventID, 129
 - cMaxStateID, 129
 - EEventID, 127
 - EStateID, 128
 - GetStateIndex, 129
- gdcmm::network::AAAbortPDU, 133
 - AAAbortPDU, 134
 - IsLastFragment, 134
 - Print, 134
 - Read, 134
 - Size, 134
 - Write, 135
- gdcmm::network::AAssociateACPDU, 135
 - AAssociateACPDU, 137
 - AAssociateRQPDU, 137
 - AddPresentationContextAC, 137
 - GetNumberOfPresentationContextAC, 137
 - GetPresentationContextAC, 137
 - GetUserInformation, 137
 - InitFromRQ, 137
 - IsLastFragment, 137
 - Print, 137
 - Read, 137
 - SetCalledAETitle, 137
 - SetCallingAETitle, 137
 - Size, 137
 - SizeType, 137
 - Write, 137
- gdcmm::network::AAssociateRJPDU, 138
 - AAssociateRJPDU, 139
 - IsLastFragment, 139
 - Print, 139
 - Read, 139
 - Size, 139
 - Write, 139
- gdcmm::network::AAssociateRQPDU, 139
 - AAssociateRQPDU, 141
 - AddPresentationContext, 141
 - GetCalledAETitle, 141
 - GetCallingAETitle, 141
 - GetNumberOfPresentationContext, 141
 - GetPresentationContext, 141
 - GetPresentationContextByAbstractSyntax, 142
 - GetPresentationContextByID, 142
 - GetPresentationContexts, 142

- IsAETitleValid, 142
- IsLastFragment, 142
- PresentationContextArrayType, 141
- Print, 142
- Read, 142
- SetCalledAETitle, 142
- SetCallingAETitle, 142
- Size, 142
- SizeType, 141
- Write, 142
- gdcmm::network::ARTIMTimer, 159
 - ARTIMTimer, 160
 - GetElapsedTime, 160
 - GetHasExpired, 160
 - GetTimeout, 160
 - SetTimeout, 160
 - Start, 160
 - Stop, 160
- gdcmm::network::AResetRPPDU, 156
 - AResetRPPDU, 157
 - IsLastFragment, 157
 - Print, 157
 - Read, 157
 - Size, 157
 - Write, 157
- gdcmm::network::AResetRQPDU, 157
 - AResetRQPDU, 158
 - IsLastFragment, 159
 - Print, 159
 - Read, 159
 - Size, 159
 - Write, 159
- gdcmm::network::AbstractSyntax, 144
 - AbstractSyntax, 144
 - GetAsDataElement, 144
 - GetName, 144
 - operator==, 144
 - Print, 144
 - Read, 144
 - SetName, 144
 - SetNameFromUID, 144
 - Size, 144
 - Write, 144
- gdcmm::network::ApplicationContext, 153
 - ApplicationContext, 153
 - GetName, 154
 - Print, 154
 - Read, 154
 - SetName, 154
 - Size, 154
 - Write, 154
- gdcmm::network::AsynchronousOperationsWindowSub, 161
 - AsynchronousOperationsWindowSub, 162
 - Read, 162
 - Size, 162
 - Write, 162
- gdcmm::network::BaseCompositeMessage, 189
 - ConstructPDV, 191
- gdcmm::network::BasePDU, 191
 - ~BasePDU, 192
 - IsLastFragment, 192
 - Print, 192
 - Read, 192
 - Size, 192
 - Write, 193
- gdcmm::network::CEchoRQ, 218
 - AffectedSOPClassUID, 220
 - ConstructPDV, 220
 - MessageID, 220
- gdcmm::network::CEchoRSP, 220
 - ConstructPDV, 221
- gdcmm::network::CFind, 221
- gdcmm::network::CFindCancelRQ, 221
 - ConstructPDV, 222
- gdcmm::network::CFindRQ, 223
 - ConstructPDV, 223
- gdcmm::network::CFindRSP, 224
 - ConstructPDV, 225
- gdcmm::network::CMoveCancelRq, 225
 - ConstructPDV, 226
- gdcmm::network::CMoveRQ, 226
 - ConstructPDV, 227
- gdcmm::network::CMoveRSP, 228
 - ConstructPDV, 229
- gdcmm::network::CStoreRQ, 260
 - ConstructPDV, 261
- gdcmm::network::CStoreRSP, 261
 - ConstructPDV, 262
- gdcmm::network::CompositeMessageFactory, 239
 - ConstructCEchoRQ, 239
 - ConstructCFindRQ, 239
 - ConstructCMoveRQ, 239
 - ConstructCStoreRQ, 239
 - ConstructCStoreRSP, 240
- gdcmm::network::DIMSE, 308
 - CommandTypes, 308
- gdcmm::network::ImplementationClassUIDSub, 430
 - ImplementationClassUIDSub, 430
 - Read, 430
 - Size, 430
 - Write, 430
- gdcmm::network::ImplementationUIDSub, 430
 - ImplementationUIDSub, 431
 - Write, 431
- gdcmm::network::ImplementationVersionNameSub, 431
 - ImplementationVersionNameSub, 431
 - Read, 431

- Size, 431
- Write, 431
- gdcmm::network::MaximumLengthSub, 474
 - GetMaximumLength, 475
 - MaximumLengthSub, 475
 - Read, 475
 - SetMaximumLength, 475
 - Size, 475
 - Write, 475
- gdcmm::network::PDUFactory, 528
 - ConstructAbortPDU, 528
 - ConstructPDU, 528
 - ConstructReleasePDU, 528
 - CreateCEchoPDU, 529
 - CreateCFindPDU, 529
 - CreateCMovePDU, 529
 - CreateCStoreRQPDU, 529
 - CreateCStoreRSPPDU, 529
 - DetermineEventByPDU, 529
 - GetPDVs, 529
- gdcmm::network::PDataTFPDU, 520
 - AddPresentationDataValue, 521
 - GetNumberOfPresentationDataValues, 521
 - GetPresentationDataValue, 521
 - IsLastFragment, 521
 - PDataTFPDU, 521
 - Print, 521
 - Read, 521
 - ReadInto, 522
 - Size, 522
 - SizeType, 521
 - Write, 522
- gdcmm::network::PresentationContextAC, 555
 - GetPresentationContextID, 555
 - GetTransferSyntax, 555
 - PresentationContextAC, 555
 - Print, 555
 - Read, 555
 - SetPresentationContextID, 555
 - SetTransferSyntax, 556
 - Size, 556
 - Write, 556
- gdcmm::network::PresentationContextRQ, 558
 - AddTransferSyntax, 559
 - GetAbstractSyntax, 559
 - GetNumberOfTransferSyntaxes, 559
 - GetPresentationContextID, 559
 - GetTransferSyntax, 559
 - GetTransferSyntaxes, 559
 - operator==, 560
 - PresentationContextRQ, 559
 - Print, 560
 - Read, 560
 - SetAbstractSyntax, 560
 - SetPresentationContextID, 560
 - Size, 560
 - SizeType, 559
 - Write, 560
- gdcmm::network::PresentationDataValue, 560
 - ConcatenatePDVBlobs, 561
 - GetBlob, 561
 - GetIsCommand, 561
 - GetIsLastFragment, 561
 - GetMessageHeader, 561
 - GetPresentationContextID, 561
 - PresentationDataValue, 561
 - Print, 561
 - Read, 561
 - ReadInto, 561
 - SetBlob, 561
 - SetCommand, 561
 - SetDataSet, 561
 - SetLastFragment, 561
 - SetMessageHeader, 561
 - SetPresentationContextID, 561
 - Size, 562
 - Write, 562
- gdcmm::network::TableRow, 694
 - transitions, 694
- gdcmm::network::TransferSyntaxSub, 712
 - GetName, 713
 - operator==, 713
 - Print, 713
 - Read, 713
 - SetName, 713
 - SetNameFromUID, 713
 - Size, 713
 - TransferSyntaxSub, 713
 - Write, 713
- gdcmm::network::Transition, 713
 - ~Transition, 714
 - mAction, 715
 - mEnd, 715
 - MakeNew, 715
 - Transition, 714
- gdcmm::network::ULAction, 738
 - ~ULAction, 740
 - PerformAction, 740
 - ULAction, 740
- gdcmm::network::ULActionAA1, 741
 - PerformAction, 741
- gdcmm::network::ULActionAA2, 742
 - PerformAction, 742
- gdcmm::network::ULActionAA3, 743
 - PerformAction, 744
- gdcmm::network::ULActionAA4, 744
 - PerformAction, 745
- gdcmm::network::ULActionAA5, 745

PerformAction, 746
gdcmm::network::ULActionAA6, 746
 PerformAction, 747
gdcmm::network::ULActionAA7, 748
 PerformAction, 748
gdcmm::network::ULActionAA8, 749
 PerformAction, 749
gdcmm::network::ULActionAE1, 750
 PerformAction, 751
gdcmm::network::ULActionAE2, 751
 PerformAction, 752
gdcmm::network::ULActionAE3, 752
 PerformAction, 753
gdcmm::network::ULActionAE4, 753
 PerformAction, 754
gdcmm::network::ULActionAE5, 755
 PerformAction, 755
gdcmm::network::ULActionAE6, 756
 PerformAction, 756
gdcmm::network::ULActionAE7, 757
 PerformAction, 758
gdcmm::network::ULActionAE8, 758
 PerformAction, 759
gdcmm::network::ULActionAR1, 759
 PerformAction, 760
gdcmm::network::ULActionAR10, 760
 PerformAction, 761
gdcmm::network::ULActionAR2, 762
 PerformAction, 762
gdcmm::network::ULActionAR3, 763
 PerformAction, 763
gdcmm::network::ULActionAR4, 764
 PerformAction, 765
gdcmm::network::ULActionAR5, 765
 PerformAction, 766
gdcmm::network::ULActionAR6, 766
 PerformAction, 767
gdcmm::network::ULActionAR7, 767
 PerformAction, 768
gdcmm::network::ULActionAR8, 769
 PerformAction, 769
gdcmm::network::ULActionAR9, 770
 PerformAction, 770
gdcmm::network::ULActionDT1, 771
 PerformAction, 772
gdcmm::network::ULActionDT2, 772
 PerformAction, 773
gdcmm::network::ULBasicCallback, 773
 ~ULBasicCallback, 774
 GetDataSets, 774
 HandleDataSet, 774
 ULBasicCallback, 774
gdcmm::network::ULConnection, 775
 ~ULConnection, 776

AddAcceptedPresentationContext, 776
FindContext, 776
GetAcceptedPresentationContexts, 776
GetConnectionInfo, 776
GetMaxPDUSize, 776
GetPresentationContextACByID, 776
GetPresentationContextIDFromPresentationContext, 776
GetPresentationContextRQByID, 776
GetPresentationContexts, 776
GetProtocol, 776
GetState, 776
GetTimer, 776
InitializeConnection, 776
InitializeIncomingConnection, 776
SetMaxPDUSize, 777
SetPresentationContexts, 777
SetState, 777
StopProtocol, 777
ULConnection, 776
gdcmm::network::ULConnectionCallback, 777
 ~ULConnectionCallback, 778
 DataSetHandled, 778
 DataSetHandles, 778
 HandleDataSet, 778
 ResetHandledDataSet, 778
 ULConnectionCallback, 778
gdcmm::network::ULConnectionInfo, 778
 GetCalledAETitle, 779
 GetCalledComputerName, 779
 GetCalledIPAddress, 779
 GetCalledIPPort, 779
 GetCallingAETitle, 779
 GetMaxPDULength, 779
 GetUserInformation, 779
 Initialize, 779
 SetMaxPDULength, 779
 ULConnectionInfo, 779
gdcmm::network::ULConnectionManager, 779
 ~ULConnectionManager, 781
 BreakConnection, 781
 BreakConnectionNow, 781
 EstablishConnection, 781
 EstablishConnectionMove, 781
 SendEcho, 782
 SendFind, 782
 SendMove, 782
 SendStore, 782
 ULConnectionManager, 781
gdcmm::network::ULEvent, 782
 ~ULEvent, 783
 GetEvent, 783
 GetPDUs, 783
 SetEvent, 783

- SetPDU, 783
- ULError, 783
- gdcmm::network::ULTransitionTable, 783
 - HandleEvent, 784
 - PrintTable, 784
 - ULTransitionTable, 783
- gdcmm::network::ULWritingCallback, 784
 - ~ULWritingCallback, 785
 - HandleDataSet, 785
 - SetDirectory, 785
 - ULWritingCallback, 785
- gdcmm::network::UserInformation, 793
 - GetMaximumLengthSub, 794
 - Print, 794
 - Read, 794
 - Size, 794
 - UserInformation, 794
 - Write, 794
- gdcmm::static_assert_test< x >, 651
- gdcmm::terminal, 129
 - Attribute, 130
 - Color, 130
 - Mode, 130
 - setAttribute, 131
 - setbgcolor, 131
 - setfgcolor, 131
 - setmode, 131
- gdcmmAAbortPDU.h, 865
- gdcmmAAAssociateACPDU.h, 865
- gdcmmAAAssociateRJPDU.h, 866
- gdcmmAAAssociateRQPDU.h, 866
- gdcmmARTIMTimer.h, 868
- gdcmmAReleaseRPPDU.h, 867
- gdcmmAReleaseRQPDU.h, 868
- gdcmmASN1.h, 868
- gdcmmAbstractSyntax.h, 866
- gdcmmAnonymizeEvent.h, 866
- gdcmmAnonymizer.h, 867
- gdcmmApplicationContext.h, 867
- gdcmmApplicationEntity.h, 867
- gdcmmAssertAlwaysMacro
 - gdcmmTrace.h, 932
- gdcmmAssertMacro
 - gdcmmTrace.h, 932
- gdcmmAsynchronousOperationsWindowSub.h, 869
- gdcmmAttribute.h, 869
- gdcmmAudioCodec.h, 869
- gdcmmBase64.h, 870
- gdcmmBaseCompositeMessage.h, 870
- gdcmmBasePDU.h, 870
- gdcmmBaseRootQuery.h, 870
- gdcmmBasicOffsetTable.h, 871
- gdcmmBitmap.h, 871
- gdcmmBitmapToBitmapFilter.h, 872
- gdcmmByteBuffer.h, 872
- gdcmmByteSwap.h, 872
- gdcmmByteSwapFilter.h, 872
- gdcmmByteValue.h, 873
- gdcmmCEchoMessages.h, 873
- gdcmmCFindMessages.h, 873
- gdcmmCMoveMessages.h, 873
- gdcmmCP246ExplicitDataElement.h, 876
- gdcmmCSAElement.h, 877
- gdcmmCSAHeader.h, 877
- gdcmmCSAHeaderDict.h, 877
- gdcmmCSAHeaderDictEntry.h, 878
- gdcmmCStoreMessages.h, 878
- gdcmmCodeString.h, 874
- gdcmmCodec.h, 874
- gdcmmCoder.h, 874
- gdcmmCommand.h, 875
- gdcmmCommandDataSet.h, 875
- gdcmmCompositeMessageFactory.h, 875
- gdcmmCompositeNetworkFunctions.h, 876
- gdcmmConstCharWrapper.h, 876
- gdcmmCryptographicMessageSyntax.h, 876
- gdcmmCurve.h, 878
- gdcmmDICODEDIR.h, 881
- gdcmmDICODEDIRGenerator.h, 882
- gdcmmDIMSE.h, 884
- gdcmmDataElement.h, 879
- gdcmmDataEvent.h, 879
- gdcmmDataSet.h, 879
- gdcmmDataSetEvent.h, 880
- gdcmmDataSetHelper.h, 880
- gdcmmDebugMacro
 - gdcmmTrace.h, 933
- gdcmmDecoder.h, 880
- gdcmmDefinedTerms.h, 880
- gdcmmDeflateStream.h, 881
- gdcmmDefs.h, 881
- gdcmmDeltaEncodingCodec.h, 881
- gdcmmDict.h, 882
- gdcmmDictConverter.h, 882
- gdcmmDictEntry.h, 883
- gdcmmDictPrinter.h, 883
- gdcmmDicts.h, 883
- gdcmmDirectionCosines.h, 884
- gdcmmDirectory.h, 884
- gdcmmDirectoryHelper.h, 885
- gdcmmDummyValueGenerator.h, 885
- gdcmmDumper.h, 885
- gdcmmElement.h, 885
- gdcmmEncapsulatedDocument.h, 886
- gdcmmEnumeratedValues.h, 886
- gdcmmErrorMacro
 - gdcmmTrace.h, 933
- gdcmmEvent.h, 887

- gdcEventMacro, 887
- gdcEventMacro
 - gdcEvent.h, 887
- gdcException.h, 888
- gdcExplicitDataElement.h, 888
- gdcExplicitImplicitDataElement.h, 888
- gdcFiducials.h, 888
- gdcFile.h, 889
- gdcFileDerivation.h, 889
- gdcFileExplicitFilter.h, 889
- gdcFileMetaInformation.h, 890
- gdcFileSet.h, 890
- gdcFilename.h, 890
- gdcFilenameGenerator.h, 890
- gdcFindPatientRootQuery.h, 891
- gdcFindStudyRootQuery.h, 891
- gdcFragment.h, 891
- gdcGlobal.h, 892
- gdcGroupDict.h, 892
- gdcIOD.h, 898
- gdcIODEntry.h, 898
- gdcIODs.h, 898
- gdcIPPSorter.h, 899
- gdcIconImage.h, 892
- gdcIconImageFilter.h, 893
- gdcIconImageGenerator.h, 893
- gdcImage.h, 893
- gdcImageApplyLookupTable.h, 894
- gdcImageChangePhotometricInterpretation.h, 894
- gdcImageChangePlanarConfiguration.h, 894
- gdcImageChangeTransferSyntax.h, 894
- gdcImageCodec.h, 895
- gdcImageConverter.h, 895
- gdcImageFragmentSplitter.h, 895
- gdcImageHelper.h, 895
- gdcImageReader.h, 896
- gdcImageToImageFilter.h, 896
- gdcImageWriter.h, 896
- gdcImplementationClassUIDSub.h, 896
- gdcImplementationUIDSub.h, 897
- gdcImplementationVersionNameSub.h, 897
- gdcImplicitDataElement.h, 897
- gdcItem.h, 899
- gdcJPEG12Codec.h, 899
- gdcJPEG16Codec.h, 900
- gdcJPEG2000Codec.h, 900
- gdcJPEG8Codec.h, 900
- gdcJPEGCodec.h, 900
- gdcJPEGLSCodec.h, 901
- gdcKAKADUCodec.h, 901
- gdcLO.h, 901
- gdcLegacyMacro.h, 901
 - GDCM_LEGACY, 901
 - GDCM_LEGACY_BODY, 901
- gdcLookupTable.h, 902
- gdcMD5.h, 903
- gdcMacro.h, 902
- gdcMacroEntry.h, 902
 - GDCMMACROENTRY_H, 902
- gdcMacros.h, 902
- gdcMaximumLengthSub.h, 903
- gdcMediaStorage.h, 903
- gdcMeshPrimitive.h, 904
- gdcModule.h, 904
- gdcModuleEntry.h, 904
- gdcModules.h, 905
- gdcMovePatientRootQuery.h, 905
- gdcMoveStudyRootQuery.h, 905
- gdcNestedModuleEntries.h, 906
- gdcNetworkEvents.h, 906
- gdcNetworkStateID.h, 907
- gdcObject.h, 907
- gdcOrientation.h, 908
- gdcOverlay.h, 908
- gdcPDBelement.h, 909
- gdcPDBHeader.h, 910
- gdcPDFCodec.h, 910
- gdcPDUFactory.h, 910
- gdcPDataTFPDU.h, 909
- gdcPNMCodec.h, 913
- gdcPVRGCodec.h, 915
- gdcParseException.h, 908
- gdcParser.h, 909
- gdcPatient.h, 909
- gdcPersonName.h, 911
- gdcPhotometricInterpretation.h, 911
- gdcPixelFormat.h, 911
- gdcPixmap.h, 912
- gdcPixmapReader.h, 912
- gdcPixmapToPixmapFilter.h, 912
- gdcPixmapWriter.h, 912
- gdcPreamble.h, 913
- gdcPresentationContext.h, 913
- gdcPresentationContextAC.h, 913
- gdcPresentationContextGenerator.h, 914
- gdcPresentationContextRQ.h, 914
- gdcPresentationDataValue.h, 914
- gdcPrinter.h, 915
- gdcPrivateTag.h, 915
- gdcProgressEvent.h, 915
- gdcPythonFilter.h, 916
- gdcQueryBase.h, 916
- gdcQueryFactory.h, 916
- gdcQueryImage.h, 917
- gdcQueryPatient.h, 917
- gdcQuerySeries.h, 917
- gdcQueryStudy.h, 918
- gdcRAWCodec.h, 918

gdcmlRLECodec.h, 919
 gdcmlReader.h, 918
 gdcmlRescaler.h, 919
 gdcmlSHA1.h, 923
 gdcmlSOPClassUIDToIOD.h, 923
 gdcmlScanner.h, 919
 gdcmlSegment.h, 920
 gdcmlSegmentHelper.h, 920
 gdcmlSegmentReader.h, 920
 gdcmlSegmentWriter.h, 921
 gdcmlSegmentedPaletteColorLookupTable.h, 920
 gdcmlSequenceOfFragments.h, 921
 gdcmlSequenceOfItems.h, 921
 gdcmlSerieHelper.h, 921
 gdcmlSeries.h, 922
 gdcmlServiceClassUser.h, 922
 gdcmlSimpleSubjectWatcher.h, 923
 gdcmlSmartPointer.h, 923
 gdcmlSorter.h, 924
 gdcmlSpacing.h, 924
 gdcmlSpectroscopy.h, 924
 gdcmlSplitMosaicFilter.h, 924
 gdcmlStaticAssert.h, 925
 GDCM_DO_JOIN, 925
 GDCM_DO_JOIN2, 925
 GDCM_JOIN, 925
 gdcmlStreamImageReader.h, 925
 gdcmlStreamImageWriter.h, 926
 gdcmlString.h, 926
 gdcmlStringFilter.h, 926
 gdcmlStudy.h, 927
 gdcmlSubject.h, 927
 gdcmlSurface.h, 927
 gdcmlSurfaceHelper.h, 927
 gdcmlSurfaceReader.h, 928
 gdcmlSurfaceWriter.h, 928
 gdcmlSwapCode.h, 928
 gdcmlSwapper.h, 928
 gdcmlSystem.h, 929
 gdcmlTable.h, 929
 gdcmlTableEntry.h, 929
 gdcmlTableReader.h, 929
 gdcmlTag.h, 930
 gdcmlTagPath.h, 930
 gdcmlTagToVR.h, 930
 gdcmlTerminal.h, 931
 gdcmlTestDriver.h, 931
 gdcmlTesting.h, 931
 gdcmlTrace.h, 932
 GDCM_FUNCTION, 932
 gdcmlAssertAlwaysMacro, 932
 gdcmlAssertMacro, 932
 gdcmlDebugMacro, 933
 gdcmlErrorMacro, 933
 gdcmlWarningMacro, 934
 gdcmlTransferSyntax.h, 934
 gdcmlTransferSyntaxSub.h, 935
 gdcmlType.h, 935
 gdcmlTypes.h, 935
 UINT32_MAX, 935
 gdcmlUIDGenerator.h, 935
 gdcmlUIDs.h, 936
 gdcmlULAction.h, 936
 gdcmlULActionAA.h, 936
 gdcmlULActionAE.h, 937
 gdcmlULActionAR.h, 937
 gdcmlULActionDT.h, 938
 gdcmlULBasicCallback.h, 938
 gdcmlULConnection.h, 938
 gdcmlULConnectionCallback.h, 938
 gdcmlULConnectionInfo.h, 939
 gdcmlULConnectionManager.h, 939
 gdcmlULEvent.h, 939
 gdcmlULTransitionTable.h, 939
 gdcmlULWritingCallback.h, 940
 gdcmlUNExplicitDataElement.h, 940
 gdcmlUNExplicitImplicitDataElement.h, 940
 gdcmlUnpacker12Bits.h, 941
 gdcmlUsage.h, 941
 gdcmlUserInformation.h, 941
 gdcmlVL.h, 943
 gdcmlVM.h, 943
 TYPETOLENGTH, 943
 gdcmlVR.h, 944
 TYPETOENCODING, 945
 VRTypeTemplateCase, 945
 gdcmlVR16ExplicitDataElement.h, 945
 gdcmlValidate.h, 941
 gdcmlValue.h, 942
 gdcmlValueIO.h, 942
 gdcmlVersion.h, 942
 gdcmlWarningMacro
 gdcmlTrace.h, 934
 gdcmlWaveform.h, 945
 gdcmlWin32.h, 945
 GDCM_EXPORT, 946
 gdcmlWriter.h, 946
 gdcmlXMLDictReader.h, 946
 gdcmlXMLPrivateDictReader.h, 946
 gdcmlanon.man, 866
 gdcmlconv.man, 876
 gdcmdiff.man, 884
 gdcmdump.man, 885
 gdcmgendir.man, 892
 gdcmimg.man, 896
 gdcminfo.man, 898
 gdcmpdf.man, 910
 gdcmrw.man, 918

- gdcmscanner.man, 920
- gdcm SCU.man, 920
- gdcm tar.man, 931
- gdcm viewer.man, 943
- GeneralECGWaveformStorage
 - gdcm::MediaStorage, 480
 - gdcm::UIDs, 727
- GeneralElectricMagneticResonanceImageStorage
 - gdcm::MediaStorage, 480
- GeneralPurposePerformedProcedureStepSOPClass
 - gdcm::UIDs, 729
- GeneralPurposeScheduledProcedureStepSOPClass
 - gdcm::UIDs, 729
- GeneralPurposeWorklistInformationModelFIND
 - gdcm::UIDs, 729
- GeneralPurposeWorklistManagementMetaSOPClass
 - gdcm::UIDs, 729
- GeneralRelevantPatientInformationQuery
 - gdcm::UIDs, 729
- Generate
 - gdcm::DICOMDIRGenerator, 295
 - gdcm::DummyValueGenerator, 316
 - gdcm::FilenameGenerator, 365
 - gdcm::IconImageGenerator, 391
 - gdcm::UIDGenerator, 718
- GenerateFromFilenames
 - gdcm::PresentationContextGenerator, 557
- GenerateFromUID
 - gdcm::PresentationContextGenerator, 557
- GenerateUUID
 - gdcm::UIDGenerator, 718
- Get
 - gdcm::ByteBuffer, 212
- GetAETitle
 - gdcm::ServiceClassUser, 630
- GetALGOType
 - gdcm::Segment, 606
- GetALGOTypeString
 - gdcm::Segment, 606
- GetAbbreviation
 - gdcm::GroupDict, 387
- GetAbstractSyntax
 - gdcm::network::PresentationContextRQ, 559
 - gdcm::PresentationContext, 554
- GetAbstractSyntaxUID
 - gdcm::BaseRootQuery, 195
 - gdcm::FindPatientRootQuery, 371
 - gdcm::FindStudyRootQuery, 373
 - gdcm::MovePatientRootQuery, 499
 - gdcm::MoveStudyRootQuery, 501
- GetAcceptedPresentationContexts
 - gdcm::network::ULConnection, 776
- GetAlgorithmFamily
 - gdcm::Surface, 674
- GetAlgorithmName
 - gdcm::Surface, 674
- GetAlgorithmVersion
 - gdcm::Surface, 674
- GetAllFilenamesFromTagToValue
 - gdcm::Scanner, 601
- GetAllTags
 - gdcm::QueryBase, 575
- GetAnatomicRegion
 - gdcm::Segment, 606
- GetAsDataElement
 - gdcm::Attribute, 164
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 171
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
 - gdcm::Element, 319
 - gdcm::Element< TVR, VM::VM1_n >, 324
 - gdcm::network::AbstractSyntax, 144
- GetAsPoints
 - gdcm::Curve, 264
- GetAsString
 - gdcm::CodeString, 234
- GetAxisOfRotation
 - gdcm::Surface, 674
- GetBasicApplicationLevelConfidentialityProfileAttributes
 - gdcm::Anonymizer, 150
- GetBitPosition
 - gdcm::Overlay, 513
- GetBitSample
 - gdcm::LookupTable, 470
- GetBitsAllocated
 - gdcm::Overlay, 513
 - gdcm::PixelFormat, 535
- GetBitsStored
 - gdcm::PixelFormat, 535
- GetBlob
 - gdcm::network::PresentationDataValue, 561
- GetBodyPart
 - itk::GDCMImageIO2, 380
- GetBuffer
 - gdcm::Bitmap, 204
 - gdcm::ByteValue, 216
 - gdcm::Overlay, 513
 - gdcm::Parser, 519
 - gdcm::SequenceOfFragments, 617
- GetBuffer2
 - gdcm::Bitmap, 204
- GetBufferAsRGBA
 - gdcm::LookupTable, 470
- GetBufferLength
 - gdcm::Bitmap, 204
 - gdcm::JPEGLSCodec, 463
 - gdcm::PNMCodec, 551

- gdcmm::RLECodec, 596
- GetBuildVersion
 - gdcmm::Version, 798
- GetByteValue
 - gdcmm::CSAElement, 249
 - gdcmm::DataElement, 269
- GetCSADataInfo
 - gdcmm::CSAHeader, 254
- GetCSAEEnd
 - gdcmm::CSAHeader, 254
- GetCSAElementByName
 - gdcmm::CSAHeader, 254
- GetCSAHeaderDict
 - gdcmm::Dicts, 307
- GetCSAHeaderDictEntry
 - gdcmm::CSAHeaderDict, 257
- GetCSAImageHeaderInfoTag
 - gdcmm::CSAHeader, 254
- GetCSASeriesHeaderInfoTag
 - gdcmm::CSAHeader, 254
- GetCTImageSeriesUIDs
 - gdcmm::DirectoryHelper, 314
- GetCWD
 - gdcmm::System, 689
- GetCalledAETitle
 - gdcmm::network::AAssociateRQPDU, 141
 - gdcmm::network::ULConnectionInfo, 779
 - gdcmm::ServiceClassUser, 630
- GetCalledComputerName
 - gdcmm::network::ULConnectionInfo, 779
- GetCalledIPAddress
 - gdcmm::network::ULConnectionInfo, 779
- GetCalledIPPort
 - gdcmm::network::ULConnectionInfo, 779
- GetCallingAETitle
 - gdcmm::network::AAssociateRQPDU, 141
 - gdcmm::network::ULConnectionInfo, 779
- GetCenterOfRotation
 - gdcmm::Surface, 674
- GetCharacterFromCurrentLocale
 - gdcmm::QueryFactory, 576
- GetCipherType
 - gdcmm::CryptographicMessageSyntax, 246
- GetColorLevel
 - vtkImageColorViewer, 838
- GetColorWindow
 - vtkImageColorViewer, 838
- GetColumns
 - gdcmm::Bitmap, 205
 - gdcmm::Overlay, 513
- GetCommand
 - gdcmm::Subject, 670
- GetConnectionInfo
 - gdcmm::network::ULConnection, 776
- GetConstructorString
 - gdcmm::Dicts, 307
- GetContourReferencedFrameOfReferenceClassUID
 - vtkRTStructSetProperties, 852
- GetContourReferencedFrameOfReferenceInstanceUID
 - vtkRTStructSetProperties, 852
- GetCryptographicMessageSyntax
 - gdcmm::Anonymizer, 150
- GetCurrentByteIndex
 - gdcmm::Parser, 519
- GetCurrentDateTime
 - gdcmm::System, 688
- GetCurrentModuleFileName
 - gdcmm::System, 688
- GetCurrentProcessFileName
 - gdcmm::System, 689
- GetCurrentResourcesDirectory
 - gdcmm::System, 689
- GetCurve
 - gdcmm::Pixmap, 540
- GetDEEnd
 - gdcmm::DataSet, 282
- GetDES
 - gdcmm::DataSet, 282
- GetData
 - gdcmm::DataEvent, 277
- GetDataElement
 - gdcmm::Bitmap, 205
 - gdcmm::DataSet, 281, 282
 - gdcmm::Item, 445
- GetDataExtraRoot
 - gdcmm::Testing, 704
- GetDataLength
 - gdcmm::DataEvent, 277
- GetDataRoot
 - gdcmm::Testing, 704
- GetDataSet
 - gdcmm::CSAHeader, 254
 - gdcmm::DataSetEvent, 286
 - gdcmm::File, 351
- GetDataSetTransferSyntax
 - gdcmm::FileMetaInformation, 359
- GetDataSets
 - gdcmm::network::ULBasicCallback, 774
- GetDataValueRepresentation
 - gdcmm::Curve, 264
- GetDebugFlag
 - gdcmm::Trace, 708
- GetDecodeLength
 - gdcmm::Base64, 189
- GetDefaultTransferSyntax
 - gdcmm::PresentationContextGenerator, 557
- GetDefs
 - gdcmm::Global, 385

- gdcmm::TableReader, 693
- GetDescription
 - gdcmm::CSAHeaderDictEntry, 258
 - gdcmm::Exception, 343
 - gdcmm::ModuleEntry, 495
 - gdcmm::Overlay, 513
- GetDescriptiveName
 - vtkGDCMImageReader, 816
 - vtkGDCMImageWriter, 820
- GetDict
 - gdcmm::XMLDictReader, 861
- GetDictEntry
 - gdcmm::Dict, 298
 - gdcmm::Dicts, 307
 - gdcmm::PrivateDict, 566
- GetDictEntryByKeyword
 - gdcmm::Dict, 298
- GetDictEntryByName
 - gdcmm::Dict, 298
- GetDictName
 - gdcmm::DictConverter, 300
- GetDictVM
 - gdcmm::Attribute, 164
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 171
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
- GetDictVR
 - gdcmm::Attribute, 164
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 171
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
- GetDicts
 - gdcmm::Global, 385
- GetDimension
 - gdcmm::Bitmap, 205
- GetDimensions
 - gdcmm::Bitmap, 205
 - gdcmm::Curve, 265
 - gdcmm::ImageCodec, 413
- GetDimensionsValue
 - gdcmm::ImageHelper, 419
- GetDimensionsValueForResolution
 - gdcmm::StreamImageReader, 655
- GetDirectionCosines
 - gdcmm::Image, 395
- GetDirectionCosinesFromDataSet
 - gdcmm::ImageHelper, 419
- GetDirectionCosinesTolerance
 - gdcmm::IPPSorter, 441
- GetDirectionCosinesValue
 - gdcmm::ImageHelper, 420
- GetDirectories
 - gdcmm::Directory, 313
- GetElapsedTime
 - gdcmm::network::ARTIMTimer, 160
- GetElement
 - gdcmm::Tag, 697
- GetElementTag
 - gdcmm::Tag, 697
- GetEncodeLength
 - gdcmm::Base64, 189
- GetErrorCode
 - gdcmm::Parser, 519
- GetErrorFlag
 - gdcmm::Trace, 708
- GetErrorString
 - gdcmm::Parser, 519
- GetEvent
 - gdcmm::network::ULEvent, 783
- GetEventName
 - gdcmm::AnonymizeEvent, 146
 - gdcmm::DataEvent, 277
 - gdcmm::DataSetEvent, 286
 - gdcmm::Event, 341
 - gdcmm::ProgressEvent, 570
- GetExtension
 - gdcmm::Filename, 363
- GetFile
 - gdcmm::Anonymizer, 150
 - gdcmm::DICOMDIRGenerator, 295
 - gdcmm::FileDerivation, 354
 - gdcmm::FileExplicitFilter, 356
 - gdcmm::IconImageFilter, 389
 - gdcmm::PythonFilter, 573
 - gdcmm::Reader, 589
 - gdcmm::SplitMosaicFilter, 650
 - gdcmm::StreamImageReader, 655
 - gdcmm::StringFilter, 667
 - gdcmm::Writer, 858
 - vtkGDCMMedicalImageProperties, 823
- GetFileExtensions
 - vtkGDCMImageReader, 816
 - vtkGDCMImageWriter, 821
- GetFileMetaInformationVersion
 - gdcmm::FileMetaInformation, 360
- GetFileName
 - gdcmm::Filename, 363
 - gdcmm::Testing, 705
 - vtkGDCMImageWriter, 821
 - vtkGDCMThreadedImageReader2, 834
- GetFileNames
 - gdcmm::Testing, 705
- GetFilename
 - gdcmm::FilenameGenerator, 365
 - gdcmm::TableReader, 693
- GetFilenameFromTagToValue

- gdcm::Scanner, 601
- GetFileNames
 - gdcm::Directory, 313
 - gdcm::FilenameGenerator, 365
 - gdcm::Scanner, 601
 - gdcm::Sorter, 645
- GetFileNamesFromSeriesUIDs
 - gdcm::DirectoryHelper, 314
- GetFiles
 - gdcm::FileSet, 367
- GetFiniteVolume
 - gdcm::Surface, 674
- GetFirstSingleSerieUIDFileSet
 - gdcm::SerieHelper, 627
- GetForcePixelSpacing
 - gdcm::ImageHelper, 420
- GetForceRescaleInterceptSlope
 - gdcm::ImageHelper, 420
- GetFormat
 - gdcm::CSAHeader, 255
- GetFragBuffer
 - gdcm::SequenceOfFragments, 617
- GetFragment
 - gdcm::SequenceOfFragments, 617
- GetFragmentSizeMax
 - gdcm::ImageFragmentSplitter, 418
- GetFrameOfReference
 - gdcm::DirectoryHelper, 314
- GetFullLength
 - gdcm::FileMetaInformation, 360
- GetGDCMDataRoot
 - vtkGDCMTesting, 830
- GetGDCMImplementationClassUID
 - gdcm::FileMetaInformation, 360
- GetGDCMImplementationVersionName
 - gdcm::FileMetaInformation, 360
- GetGDCMSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, 360
- GetGDCMUID
 - gdcm::UIDGenerator, 718
- GetGroup
 - gdcm::Curve, 265
 - gdcm::Overlay, 513
 - gdcm::Tag, 697
- GetHasExpired
 - gdcm::network::ARTIMTimer, 160
- GetHeader
 - gdcm::File, 351
- GetHeaderInfo
 - gdcm::ImageCodec, 413
 - gdcm::JPEG12Codec, 449
 - gdcm::JPEG16Codec, 451
 - gdcm::JPEG2000Codec, 454
 - gdcm::JPEG8Codec, 456
 - gdcm::JPEGCodec, 459
 - gdcm::JPEGLSCodec, 463
 - gdcm::PNMCodec, 551
 - gdcm::RAWCodec, 586
 - gdcm::RLECodec, 596
- GetHighBit
 - gdcm::PixelFormat, 535
- GetHostName
 - gdcm::System, 689
- GetIE
 - gdcm::IODEntry, 437
- GetIOD
 - gdcm::IODs, 439
 - gdcm::SOPClassUIDToIOD, 642
- GetIODEntry
 - gdcm::IOD, 436
- GetIODFromFile
 - gdcm::Defs, 290
- GetIODFromSOPClassUID
 - gdcm::SOPClassUIDToIOD, 642
- GetIODNameFromMediaStorage
 - gdcm::Defs, 290
- GetIODs
 - gdcm::Defs, 290, 291
- GetIconImage
 - gdcm::IconImageFilter, 389
 - gdcm::IconImageGenerator, 391
 - gdcm::Pixmap, 540
 - vtkGDCMImageReader, 816
- GetImage
 - gdcm::ImageReader, 424
 - gdcm::ImageWriter, 429
 - gdcm::PixmapWriter, 548
 - gdcm::SplitMosaicFilter, 650
- GetImplementationClassUID
 - gdcm::FileMetaInformation, 360
- GetImplementationVersionName
 - gdcm::FileMetaInformation, 360
- GetIndex
 - gdcm::SwapCode, 685
 - gdcm::VM, 804
- GetInput
 - gdcm::ImageToImageFilter, 427
 - gdcm::PixmapToPixmapFilter, 546
 - vtkImageColorViewer, 839
- GetInputFilename
 - gdcm::DictConverter, 300
- GetInstance
 - gdcm::Global, 385
- GetInstitution
 - itk::GDCMImageIO2, 380
- GetIntercept
 - gdcm::Image, 395
 - gdcm::Rescaler, 593

- GetInterfile
 - gdcm::CSAHeader, 255
- GetInternal
 - gdcm::Preamble, 553
- GetIsCommand
 - gdcm::network::PresentationDataValue, 561
- GetIsLastFragment
 - gdcm::network::PresentationDataValue, 561
- GetItem
 - gdcm::SequenceOfItems, 622
- GetKey
 - gdcm::CSAElement, 249
- GetKeys
 - gdcm::Scanner, 601
- GetKeyword
 - gdcm::DictEntry, 302
- GetKeywordFromTag
 - gdcm::Dict, 298
- GetLUT
 - gdcm::Bitmap, 205
 - gdcm::ImageCodec, 413
 - gdcm::ImageHelper, 420
 - gdcm::LookupTable, 470
- GetLUTDescriptor
 - gdcm::LookupTable, 470
- GetLUTLength
 - gdcm::LookupTable, 470
- GetLabel
 - gdcm::Orientation, 509
- GetLabelFromTag
 - itk::GDCMImageIO2, 380
- GetLastElement
 - gdcm::ParseException, 517
- GetLastSystemError
 - gdcm::System, 689
- GetLength
 - gdcm::ByteValue, 216
 - gdcm::CP246ExplicitDataElement, 245
 - gdcm::DataElement, 269
 - gdcm::DataSet, 282
 - gdcm::Element, 319
 - gdcm::Element< TVR, VM::VM1_n >, 324
 - gdcm::Element< VR::AS, VM::VM5 >, 332
 - gdcm::ExplicitDataElement, 346
 - gdcm::ExplicitImplicitDataElement, 348
 - gdcm::Fragment, 376
 - gdcm::ImplicitDataElement, 433
 - gdcm::Item, 445
 - gdcm::Preamble, 553
 - gdcm::SequenceOfFragments, 618
 - gdcm::SequenceOfItems, 623
 - gdcm::Tag, 698
 - gdcm::UNExplicitDataElement, 787
 - gdcm::UNExplicitImplicitDataElement, 789
 - gdcm::Value, 797
 - gdcm::VL, 800
 - gdcm::VM, 804
 - gdcm::VR, 808, 809
 - gdcm::VR16ExplicitDataElement, 811
- GetLoadPrivateTagsDefault
 - itk::GDCMImageIO2, 380
- GetLoadSequencesDefault
 - itk::GDCMImageIO2, 380
- GetLocaleCharSet
 - gdcm::System, 689
- GetLossless
 - gdcm::JPEGCodec, 459
 - gdcm::JPEGLSCodec, 463
- GetLossyFlag
 - gdcm::ImageCodec, 413
- GetLossyFlagFromFile
 - gdcm::Testing, 705
- GetMD5DataImage
 - gdcm::Testing, 705
- GetMD5DataImages
 - gdcm::Testing, 705
- GetMD5FromBrokenFile
 - gdcm::Testing, 705
- GetMD5FromFile
 - gdcm::Testing, 705
- GetMD5MetalImage
 - vtkGDCMTesting, 830
- GetMHDMD5FromFile
 - vtkGDCMTesting, 830
- GetMPType
 - gdcm::MeshPrimitive, 489
- GetMPTypeString
 - gdcm::MeshPrimitive, 489
- GetMRImageSeriesUIDs
 - gdcm::DirectoryHelper, 315
- GetMSString
 - gdcm::MediaStorage, 481
- GetMSType
 - gdcm::MediaStorage, 481
- GetMTime
 - vtkImageMapToColors16, 843
- GetMacro
 - gdcm::Macros, 474
- GetMacroEntry
 - gdcm::Macro, 473
- GetMacros
 - gdcm::Defs, 291
- GetMajorAxisFromPatientRelativeDirectionCosine
 - gdcm::Orientation, 509
- GetMajorVersion
 - gdcm::Version, 799
- GetManifold
 - gdcm::Surface, 674

- GetManufacturer
 - itk::GDCMImageIO2, 380
- GetMapping
 - gdcm::Scanner, 602
- GetMappingFromTagToValue
 - gdcm::Scanner, 602
- GetMappings
 - gdcm::Scanner, 602
- GetMax
 - gdcm::PixelFormat, 536
- GetMaxLength
 - gdcm::PersonName, 530
- GetMaxPDULength
 - gdcm::network::ULConnectionInfo, 779
- GetMaxPDUSize
 - gdcm::network::ULConnection, 776
- GetMaximumLength
 - gdcm::network::MaximumLengthSub, 475
- GetMaximumLengthSub
 - gdcm::network::UserInformation, 794
- GetMaximumPointDistance
 - gdcm::Surface, 674
- GetMeanPointDistance
 - gdcm::Surface, 674
- GetMediaStorage
 - gdcm::FileMetaInformation, 360
- GetMediaStorageDataFile
 - gdcm::Testing, 705
- GetMediaStorageDataFiles
 - gdcm::Testing, 705
- GetMediaStorageFromFile
 - gdcm::Testing, 705
- GetMeshPrimitive
 - gdcm::Surface, 674, 675
- GetMessageHeader
 - gdcm::network::PresentationDataValue, 561
- GetMetaInformationTS
 - gdcm::FileMetaInformation, 360
- GetMin
 - gdcm::PixelFormat, 536
- GetMinorVersion
 - gdcm::Version, 799
- GetModality
 - gdcm::MediaStorage, 481
 - itk::GDCMImageIO2, 380
- GetModalityDimension
 - gdcm::MediaStorage, 481
- GetModel
 - itk::GDCMImageIO2, 380
- GetModule
 - gdcm::Modules, 497
- GetModuleEntry
 - gdcm::NestedModuleEntries, 504
- GetModuleEntryInMacros
 - gdcm::Module, 492
- GetModules
 - gdcm::Defs, 291
- GetName
 - gdcm::CSAElement, 249
 - gdcm::CSAHeaderDictEntry, 258
 - gdcm::DictEntry, 302
 - gdcm::Filename, 363
 - gdcm::GroupDict, 387
 - gdcm::IODEntry, 437
 - gdcm::Macro, 473
 - gdcm::Module, 492
 - gdcm::ModuleEntry, 495
 - gdcm::network::AbstractSyntax, 144
 - gdcm::network::ApplicationContext, 154
 - gdcm::network::TransferSyntaxSub, 713
 - gdcm::PDBElement, 523
 - gdcm::QueryBase, 575
 - gdcm::QueryImage, 578
 - gdcm::QueryPatient, 579
 - gdcm::QuerySeries, 581
 - gdcm::QueryStudy, 583
 - gdcm::UIDs, 737
- GetNeedByteSwap
 - gdcm::Bitmap, 205
 - gdcm::ImageCodec, 413
- GetNegotiatedType
 - gdcm::TransferSyntax, 711
- GetNestedDataSet
 - gdcm::Item, 445, 446
- GetNextSingleSerieUIDFileSet
 - gdcm::SerieHelper, 627
- GetNoOfItems
 - gdcm::CSAElement, 249
- GetNumberOfComponents
 - gdcm::PersonName, 530
- GetNumberOfContourReferencedFrameOfReferences
 - vtkRTStructSetProperties, 852
- GetNumberOfCurves
 - gdcm::Curve, 265
 - gdcm::Pixmap, 540
- GetNumberOfDimensions
 - gdcm::Bitmap, 205
 - gdcm::ImageCodec, 413
- GetNumberOfElementsFromArray
 - gdcm::VM, 804
- GetNumberOfFileNames
 - gdcm::Testing, 705
- GetNumberOfFileNames
 - gdcm::FilenameGenerator, 365
- GetNumberOfFragments
 - gdcm::SequenceOfFragments, 618
- GetNumberOfIODs
 - gdcm::IOD, 436

- GetNumberOfIconImages
 - gdcm::IconImageFilter, 389
- GetNumberOfItems
 - gdcm::SequenceOfItems, 623
- GetNumberOfMD5DataImages
 - gdcm::Testing, 705
- GetNumberOfMD5MetaImages
 - vtkGDCMTesting, 830
- GetNumberOfMSSString
 - gdcm::MediaStorage, 482
- GetNumberOfMSType
 - gdcm::MediaStorage, 482
- GetNumberOfMediaStorageDataFiles
 - gdcm::Testing, 705
- GetNumberOfModality
 - gdcm::MediaStorage, 482
- GetNumberOfModuleEntries
 - gdcm::NestedModuleEntries, 504
- GetNumberOfOverlays
 - gdcm::Pixmap, 540
- GetNumberOfPoints
 - gdcm::Curve, 265
- GetNumberOfPresentationContext
 - gdcm::network::AAssociateRQPDU, 141
- GetNumberOfPresentationContextAC
 - gdcm::network::AAssociateACPDU, 137
- GetNumberOfPresentationDataValues
 - gdcm::network::PDDataTFPDU, 521
- GetNumberOfPrimitivesData
 - gdcm::MeshPrimitive, 489
- GetNumberOfReferencedFrameOfReferences
 - vtkRTStructSetProperties, 852
- GetNumberOfSOPClassToIOD
 - gdcm::SOPClassUIDToIOD, 643
- GetNumberOfSegments
 - gdcm::SegmentWriter, 614
- GetNumberOfSeriesInStudy
 - itk::GDCMImageIO2, 380
- GetNumberOfStructureSetROIs
 - vtkRTStructSetProperties, 852
- GetNumberOfStudyRelatedSeries
 - itk::GDCMImageIO2, 380
- GetNumberOfSurfacePoints
 - gdcm::Surface, 675
- GetNumberOfSurfaces
 - gdcm::SurfaceReader, 681
 - gdcm::SurfaceWriter, 683
- GetNumberOfTransferSyntaxStrings
 - gdcm::UIDs, 737
- GetNumberOfTransferSyntaxes
 - gdcm::network::PresentationContextRQ, 559
 - gdcm::PresentationContext, 554
- GetNumberOfValues
 - gdcm::Attribute, 165
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 171
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
- GetNumberOfVectors
 - gdcm::Surface, 675
- GetObliquityThresholdCosineValue
 - gdcm::Orientation, 509
- GetOffScreenRendering
 - vtkImageColorViewer, 839
- GetOptionalTags
 - gdcm::QueryBase, 575
 - gdcm::QueryImage, 578
 - gdcm::QueryPatient, 579
 - gdcm::QuerySeries, 581
 - gdcm::QueryStudy, 583
- GetOrderedValues
 - gdcm::Scanner, 602
- GetOrigin
 - gdcm::Image, 395
 - gdcm::Overlay, 513
- GetOriginValue
 - gdcm::ImageHelper, 420
- GetOutput
 - gdcm::ImageConverter, 416
- GetOutput
 - gdcm::BitmapToBitmapFilter, 211
 - gdcm::ImageToImageFilter, 427
 - gdcm::PixmapToPixmapFilter, 546
- GetOutputFilename
 - gdcm::DictConverter, 300
- GetOutputType
 - gdcm::DictConverter, 300
- GetOverlay
 - gdcm::Pixmap, 540
 - vtkGDCMImageReader, 816
- GetOverlayData
 - gdcm::Overlay, 513
- GetOverlayVisibility
 - vtkImageColorViewer, 839
- GetOwner
 - gdcm::PrivateTag, 568
- GetPDBEEnd
 - gdcm::PDBHeader, 525
- GetPDBElementByName
 - gdcm::PDBHeader, 525
- GetPDBInfoTag
 - gdcm::PDBHeader, 525
- GetPDUs
 - gdcm::network::ULEvent, 783
- GetPDVs
 - gdcm::network::PDUFactory, 529
- GetPIString
 - gdcm::PhotometricInterpretation, 532

- GetPIType
 - gdcm::PhotometricInterpretation, 532
- GetPath
 - gdcm::Filename, 363
- GetPatientAge
 - itk::GDCMImageIO2, 380
- GetPatientDOB
 - itk::GDCMImageIO2, 380
- GetPatientID
 - itk::GDCMImageIO2, 380
- GetPatientName
 - itk::GDCMImageIO2, 380
- GetPatientSex
 - itk::GDCMImageIO2, 381
- GetPattern
 - gdcm::FilenameGenerator, 366
- GetPermissions
 - gdcm::System, 689
- GetPhotometricInterpretation
 - gdcm::Bitmap, 205
 - gdcm::ImageChangePhotometricInterpretation, 402
 - gdcm::ImageCodec, 413
- GetPhotometricInterpretationValue
 - gdcm::ImageHelper, 420
- GetPixelFormat
 - gdcm::Bitmap, 205, 206
 - gdcm::ImageCodec, 413
- GetPixelFormatValue
 - gdcm::ImageHelper, 420
- GetPixelRepresentation
 - gdcm::PixelFormat, 536
- GetPixelSize
 - gdcm::PixelFormat, 536
- GetPixelSpacingDataRoot
 - gdcm::Testing, 706
- GetPixmap
 - gdcm::IconImageGenerator, 392
 - gdcm::PixmapReader, 543
 - gdcm::PixmapWriter, 548, 549
- GetPlanarConfiguration
 - gdcm::Bitmap, 206
 - gdcm::ImageChangePlanarConfiguration, 405
 - gdcm::ImageCodec, 413
- GetPlanarConfigurationValue
 - gdcm::ImageHelper, 420
- GetPointCoordinatesData
 - gdcm::Surface, 675
- GetPointPositionAccuracy
 - gdcm::Surface, 675
- GetPointer
 - gdcm::ByteValue, 217
 - gdcm::LookupTable, 470
 - gdcm::SmartPointer, 641
 - vtkLookupTable16, 849
- GetPointerFromElement
 - gdcm::ImageHelper, 420
- GetPointsBoundingBoxCoordinates
 - gdcm::Surface, 675
- GetPosition
 - vtkImageColorViewer, 839
- GetPreamble
 - gdcm::FileMetaInformation, 360
- GetPrefix
 - gdcm::FilenameGenerator, 366
- GetPresentationContext
 - gdcm::network::AAssociateRQPDU, 141
- GetPresentationContextAC
 - gdcm::network::AAssociateACPDU, 137
- GetPresentationContextACByID
 - gdcm::network::ULConnection, 776
- GetPresentationContextByAbstractSyntax
 - gdcm::network::AAssociateRQPDU, 142
- GetPresentationContextByID
 - gdcm::network::AAssociateRQPDU, 142
- GetPresentationContextID
 - gdcm::network::PresentationContextAC, 555
 - gdcm::network::PresentationContextRQ, 559
 - gdcm::network::PresentationDataValue, 561
 - gdcm::PresentationContext, 554
- GetPresentationContextIDFromPresentationContext
 - gdcm::network::ULConnection, 776
- GetPresentationContextRQByID
 - gdcm::network::ULConnection, 776
- GetPresentationContexts
 - gdcm::network::AAssociateRQPDU, 142
 - gdcm::network::ULConnection, 776
 - gdcm::PresentationContextGenerator, 558
- GetPresentationDataValue
 - gdcm::network::PDataTFPDU, 521
- GetPrimitiveData
 - gdcm::MeshPrimitive, 489
- GetPrimitiveType
 - gdcm::MeshPrimitive, 490
- GetPrimitivesData
 - gdcm::MeshPrimitive, 489
- GetPrintStyle
 - gdcm::Printer, 564
- GetPrivateCreator
 - gdcm::DataSet, 282
 - gdcm::Tag, 698
- GetPrivateDict
 - gdcm::Dicts, 307
 - gdcm::XMLPrivateDictReader, 863
- GetProcessingAlgorithm
 - gdcm::Surface, 675
- GetProgress
 - gdcm::ProgressEvent, 570
- GetPropertyCategory

- gdcm::Segment, 606
- GetPropertyType
 - gdcm::Segment, 606
- GetProtocol
 - gdcm::network::ULConnection, 776
- GetPublicDict
 - gdcm::Dicts, 307
- GetQuality
 - gdcm::JPEG2000Codec, 454
 - gdcm::JPEGCodec, 459
- GetQueryDataSet
 - gdcm::BaseRootQuery, 195
- GetQueryLevel
 - gdcm::QueryBase, 575
 - gdcm::QueryImage, 578
 - gdcm::QueryPatient, 580
 - gdcm::QuerySeries, 581
 - gdcm::QueryStudy, 583
- GetRAWMD5FromFile
 - vtkGDCMTesting, 830
- GetRTStructSeriesUIDs
 - gdcm::DirectoryHelper, 315
- GetRate
 - gdcm::JPEG2000Codec, 454
- GetRecommendedDisplayCIELabValue
 - gdcm::Surface, 675
- GetRecommendedDisplayGrayscaleValue
 - gdcm::Surface, 675
- GetRecommendedPresentationOpacity
 - gdcm::Surface, 675
- GetRecommendedPresentationType
 - gdcm::Surface, 675
- GetRef
 - gdcm::IODEntry, 437
- GetReferencedFrameOfReferenceClassUID
 - vtkRTStructSetProperties, 852
- GetReferencedFrameOfReferenceInstanceUID
 - vtkRTStructSetProperties, 852
- GetRequiredTags
 - gdcm::QueryBase, 575
 - gdcm::QueryImage, 578
 - gdcm::QueryPatient, 580
 - gdcm::QuerySeries, 581
 - gdcm::QueryStudy, 583
- GetRescaleInterceptSlopeValue
 - gdcm::ImageHelper, 420
- GetRetired
 - gdcm::DictEntry, 302
- GetRoot
 - gdcm::UIDGenerator, 718
- GetRows
 - gdcm::Bitmap, 206
 - gdcm::Overlay, 513
- GetSOPClassUID
 - gdcm::DirectoryHelper, 315
 - GetSOPClassUIDFromIOD
 - gdcm::SOPClassUIDToIOD, 643
 - GetSOPClassUIDToIOD
 - gdcm::SOPClassUIDToIOD, 643
 - GetSOPClassUIDToIODs
 - gdcm::SOPClassUIDToIOD, 643
 - GetSTATES
 - gdcm::Surface, 675
 - GetSTATESString
 - gdcm::Surface, 675
 - GetSamplesPerPixel
 - gdcm::PhotometricInterpretation, 532
 - gdcm::PixelFormat, 536
 - GetScalarType
 - gdcm::PixelFormat, 536
 - GetScalarTypeAsString
 - gdcm::PixelFormat, 536
 - GetScanOptions
 - itk::GDCMImageIO2, 381
 - GetScanner
 - gdcm::DICOMDIRGenerator, 296
 - GetSegment
 - gdcm::SegmentWriter, 614
 - GetSegmentAlgorithmName
 - gdcm::Segment, 606
 - GetSegmentAlgorithmType
 - gdcm::Segment, 606
 - GetSegmentDescription
 - gdcm::Segment, 606
 - GetSegmentLabel
 - gdcm::Segment, 606
 - GetSegmentNumber
 - gdcm::Segment, 606
 - GetSegments
 - gdcm::SegmentReader, 611
 - gdcm::SegmentWriter, 614
 - GetSelectedTagsOffsetFromFile
 - gdcm::Testing, 706
 - GetSequenceOfFragments
 - gdcm::DataElement, 269
 - GetSequenceOfItems
 - gdcm::DataElement, 269, 270
 - GetSeriesUIDsBySOPClassUID
 - gdcm::DirectoryHelper, 315
 - GetSize
 - gdcm::VR, 809
 - vtkImageColorViewer, 839
 - GetSizeof
 - gdcm::VR, 809
 - GetSliceMax
 - vtkImageColorViewer, 839
 - GetSliceMin
 - vtkImageColorViewer, 839

- GetSliceRange
 - vtkImageColorViewer, 839
- GetSlope
 - gdcm::Image, 396
 - gdcm::Rescaler, 593
- GetSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, 360
- GetSourceDirectory
 - gdcm::Testing, 706
- GetSpacing
 - gdcm::Image, 396
- GetSpacingTagFromMediaStorage
 - gdcm::ImageHelper, 420
- GetSpacingValue
 - gdcm::ImageHelper, 420
- GetStart
 - gdcm::ByteBuffer, 212
- GetState
 - gdcm::network::ULConnection, 776
- GetStateIndex
 - gdcm::network, 129
- GetStream
 - gdcm::Trace, 708
- GetStreamOffsetFromFile
 - gdcm::Testing, 706
- GetStreamPtr
 - gdcm::Reader, 589
 - gdcm::Writer, 858
- GetString
 - gdcm::MediaStorage, 482
 - gdcm::PhotometricInterpretation, 532
 - gdcm::TransferSyntax, 711
 - gdcm::UIDs, 738
- GetStructureSetObservationNumber
 - vtkRTStructSetProperties, 852
- GetStructureSetROIGenerationAlgorithm
 - vtkRTStructSetProperties, 852
- GetStructureSetROIName
 - vtkRTStructSetProperties, 852
- GetStructureSetROINumber
 - vtkRTStructSetProperties, 852
- GetStructureSetROIRefFrameRefUID
 - vtkRTStructSetProperties, 852
- GetStructureSetRTROIInterpretedType
 - vtkRTStructSetProperties, 852
- GetStudyDate
 - itk::GDCMImageIO2, 381
- GetStudyDescription
 - itk::GDCMImageIO2, 381
- GetStudyID
 - itk::GDCMImageIO2, 381
- GetSurface
 - gdcm::Segment, 606
- GetSurfaceComments
 - gdcm::Surface, 675
- GetSurfaceCount
 - gdcm::Segment, 606
- GetSurfaceNumber
 - gdcm::Surface, 675
- GetSurfaceProcessing
 - gdcm::Surface, 675
- GetSurfaceProcessingDescription
 - gdcm::Surface, 675
- GetSurfaceProcessingRatio
 - gdcm::Surface, 675
- GetSurfaces
 - gdcm::Segment, 607
- GetSwapCode
 - gdcm::TransferSyntax, 711
- GetSwapCodeString
 - gdcm::SwapCode, 685
- GetSyngoDT
 - gdcm::CSAElement, 249
- GetTSString
 - gdcm::TransferSyntax, 711
- GetTSType
 - gdcm::TransferSyntax, 711
- GetTable
 - gdcm::SequenceOfFragments, 618
- GetTableEntry
 - gdcm::Table, 691
- GetTag
 - gdcm::AnonymizeEvent, 146
 - gdcm::Attribute, 165
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 171
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
 - gdcm::DataElement, 270
- GetTagListByLevel
 - gdcm::BaseRootQuery, 195
 - gdcm::FindPatientRootQuery, 371
 - gdcm::FindStudyRootQuery, 373
 - gdcm::MovePatientRootQuery, 499
 - gdcm::MoveStudyRootQuery, 501
- GetTempDirectory
 - gdcm::Testing, 706
- GetTempDirectoryW
 - gdcm::Testing, 706
- GetTempFilename
 - gdcm::Testing, 706
- GetTempFilenameW
 - gdcm::Testing, 706
- GetTimeout
 - gdcm::network::ARTIMTimer, 160
 - gdcm::ServiceClassUser, 630
- GetTimer
 - gdcm::network::ULConnection, 776

- GetTimezoneOffsetFromUTC
 - gdcm::System, 689
- GetToplevel
 - gdcm::Directory, 313
- GetTransferSyntax
 - gdcm::Bitmap, 206
 - gdcm::ImageChangeTransferSyntax, 408
 - gdcm::network::PresentationContextAC, 555
 - gdcm::network::PresentationContextRQ, 559
 - gdcm::PresentationContext, 554
- GetTransferSyntaxString
 - gdcm::UIDs, 738
- GetTransferSyntaxStrings
 - gdcm::UIDs, 738
- GetTransferSyntaxes
 - gdcm::network::PresentationContextRQ, 559
- GetType
 - gdcm::ModuleEntry, 495
 - gdcm::Orientation, 509
 - gdcm::Overlay, 513
 - gdcm::PhotometricInterpretation, 532
- GetTypeFromTag
 - gdcm::Defs, 291
 - gdcm::IOD, 436
- GetTypeOfData
 - gdcm::Curve, 265
- GetTypeOfDataDescription
 - gdcm::Curve, 265
- GetTypeString
 - gdcm::Type, 716
- GetTypeType
 - gdcm::Type, 716
- GetUIDName
 - gdcm::UIDs, 738
- GetUIDString
 - gdcm::UIDs, 738
- GetUniqueTags
 - gdcm::QueryBase, 575
 - gdcm::QueryImage, 578
 - gdcm::QueryPatient, 580
 - gdcm::QuerySeries, 581
 - gdcm::QueryStudy, 583
- GetUnpackBuffer
 - gdcm::Overlay, 514
- GetUsage
 - gdcm::IODEntry, 437
- GetUsageString
 - gdcm::Usage, 791
- GetUsageType
 - gdcm::IODEntry, 438
 - gdcm::Usage, 791
- GetUserData
 - gdcm::Parser, 519
- GetUserInformation
 - gdcm::network::AAAssociateACPDU, 137
 - gdcm::network::ULConnectionInfo, 779
- GetVIEWType
 - gdcm::Surface, 676
- GetVIEWTypeString
 - gdcm::Surface, 676
- GetVL
 - gdcm::DataElement, 271
- GetVL16Max
 - gdcm::VL, 800
- GetVL32Max
 - gdcm::VL, 800
- GetVM
 - gdcm::Attribute, 166
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 171
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 174
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 176
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_-2n >, 181
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 183
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_-3n >, 184
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 186
 - gdcm::CSAElement, 250
 - gdcm::CSAHeaderDictEntry, 258
 - gdcm::DictEntry, 302
 - gdcm::Element, 320
 - gdcm::Element< TVR, VM::VM1_n >, 324
- GetVMString
 - gdcm::VM, 804
- GetVMType
 - gdcm::VM, 804
- GetVMTypeFromLength
 - gdcm::VM, 805
- GetVR
 - gdcm::Attribute, 166
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
 - gdcm::CSAElement, 250
 - gdcm::CSAHeaderDictEntry, 258
 - gdcm::DataElement, 271
 - gdcm::DictEntry, 302
 - gdcm::Element, 320
 - gdcm::Element< TVR, VM::VM1_n >, 324
- GetVRFromTag

- gdcmm, 119
- GetVRString
 - gdcmm::VR, 809
- GetVRStringFromFile
 - gdcmm::VR, 809
- GetVRType
 - gdcmm::VR, 809
- GetVRTypeFromFile
 - gdcmm::VR, 809
- GetVTKDataRoot
 - vtkGDCMTesting, 830
- GetValidatedFile
 - gdcmm::Validate, 795
- GetValue
 - gdcmm::Attribute, 165
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 171
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
 - gdcmm::CSAElement, 249
 - gdcmm::DataElement, 270
 - gdcmm::Element, 319
 - gdcmm::Element< TVR, VM::VM1_n >, 324
 - gdcmm::PDBelement, 523
 - gdcmm::Scanner, 602
- GetValueAsSQ
 - gdcmm::DataElement, 270
- GetValueFromTag
 - itk::GDCMImageIO2, 381
- GetValues
 - gdcmm::Attribute, 165
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 171
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 178
 - gdcmm::Element, 320
 - gdcmm::Scanner, 602
- GetVectorAccuracy
 - gdcmm::Surface, 675
- GetVectorCoordinateData
 - gdcmm::Surface, 676
- GetVectorDimensionality
 - gdcmm::Surface, 676
- GetVersion
 - gdcmm::Version, 799
- GetWarningFlag
 - gdcmm::Trace, 708
- GetWindowName
 - vtkImageColorViewer, 839
- GetZSpacing
 - gdcmm::IPPSorter, 441
- GetZSpacingTagFromMediaStorage
 - gdcmm::ImageHelper, 420
- GetZSpacingTolerance
 - gdcmm::IPPSorter, 442
- Global
 - gdcmm::Defs, 291
 - gdcmm::Dicts, 307
 - gdcmm::Global, 385
- GlobalInstance
 - gdcmm, 123
- GrabOverlayFromPixelData
 - gdcmm::Overlay, 514
- GrayscaleSoftcopyPresentationStateStorageSOPClass
 - gdcmm::MediaStorage, 480
 - gdcmm::UIDs, 728
- green
 - gdcmm::terminal, 130
- group
 - gdcmm::SerieHelper::Rule, 597
- GroupDict
 - gdcmm::GroupDict, 387
- GroupStringVector
 - gdcmm::GroupDict, 387
- GuessFromModality
 - gdcmm::MediaStorage, 482
- HSV
 - gdcmm::PhotometricInterpretation, 532
- HandleDataSet
 - gdcmm::network::ULBasicCallback, 774
 - gdcmm::network::ULConnectionCallback, 778
 - gdcmm::network::ULWritingCallback, 785
- HandleDescription
 - gdcmm::XMLDictReader, 861
 - gdcmm::XMLPrivateDictReader, 863
- HandleEntry
 - gdcmm::XMLDictReader, 861
 - gdcmm::XMLPrivateDictReader, 863
- HandleEvent
 - gdcmm::network::ULTransitionTable, 784
- HandleIOD
 - gdcmm::TableReader, 693
- HandleIODEntry
 - gdcmm::TableReader, 693
- HandleMacro
 - gdcmm::TableReader, 693
- HandleMacroEntry
 - gdcmm::TableReader, 693
- HandleMacroEntryDescription
 - gdcmm::TableReader, 693
- HandleModule
 - gdcmm::TableReader, 693
- HandleModuleEntry
 - gdcmm::TableReader, 693
- HandleModuleEntryDescription
 - gdcmm::TableReader, 693
- HandleModuleInclude

- gdcM::TableReader, 694
- HangingProtocolInformationModelFIND
 - gdcM::UIDs, 730
- HangingProtocolInformationModelMOVE
 - gdcM::UIDs, 730
- HangingProtocolStorage
 - gdcM::MediaStorage, 480
 - gdcM::UIDs, 730
- HardcopyColorImageStorageSOPClassRetired
 - gdcM::UIDs, 727
- HardcopyGrayscaleImageStorage
 - gdcM::MediaStorage, 480
- HardcopyGrayscaleImageStorageSOPClassRetired
 - gdcM::UIDs, 727
- HasObserver
 - gdcM::Subject, 670
- HemodynamicWaveformStorage
 - gdcM::MediaStorage, 480
 - gdcM::UIDs, 727
- hidden
 - gdcM::terminal, 130
- ICBM452T1FrameofReference
 - gdcM::UIDs, 726
- ICBMSingleSubjectMRIFrameofReference
 - gdcM::UIDs, 726
- INT12
 - gdcM::PixelFormat, 535
- INT16
 - gdcM::PixelFormat, 535
- INT32
 - gdcM::PixelFormat, 535
- INT8
 - gdcM::PixelFormat, 535
- INTERFILE
 - gdcM::CSAHeader, 253
- INVALID
 - gdcM::VR, 807
- IS
 - gdcM::VR, 808
- IOD
 - gdcM::IOD, 435
- IODEntry
 - gdcM::IODEntry, 437
- IODMapType
 - gdcM::IODs, 439
- IODMapTypeConstIterator
 - gdcM::IODs, 439
- IODName
 - gdcM::IODs, 439
- IODs
 - gdcM::IODs, 439
- IPPSorter
 - gdcM::IPPSorter, 441
- ITK_GDCM_EXPORT
 - itkGDCMImageIO2.h, 947
- Icon
 - gdcM::Pixmap, 541
- IconDataScalarType
 - vtkGDCMImageReader, 818
- IconImage
 - gdcM, 117
- IconImageDataExtent
 - vtkGDCMImageReader, 818
- IconImageFilter
 - gdcM::IconImageFilter, 389
- IconImageGenerator
 - gdcM::IconImageGenerator, 391
- IconNumberOfScalarComponents
 - vtkGDCMImageReader, 818
- ignore_char
 - gdcM::ignore_char, 393
- Image
 - gdcM::Image, 395
- ImageOverlayBoxSOPClassRetired
 - gdcM::UIDs, 727
- ImageActor
 - vtkImageColorViewer, 841
- ImageApplyLookupTable
 - gdcM::ImageApplyLookupTable, 399
- ImageChangePhotometricInterpretation
 - gdcM::ImageChangePhotometricInterpretation, 402
 - gdcM::ImageCodec, 414
- ImageChangePlanarConfiguration
 - gdcM::ImageChangePlanarConfiguration, 405
- ImageChangeTransferSyntax
 - gdcM::Bitmap, 208
 - gdcM::ImageChangeTransferSyntax, 408
- ImageCodec
 - gdcM::ImageCodec, 412
- ImageConverter
 - gdcM::ImageConverter, 416
- ImageFormat
 - vtkGDCMImageReader, 818
- ImageFragmentSplitter
 - gdcM::ImageFragmentSplitter, 418
- ImageOrientationPatient
 - vtkGDCMImageReader, 818
- ImagePositionPatient
 - vtkGDCMImageReader, 818
- ImagePositionPatientOrdering
 - gdcM::SerieHelper, 627
- ImageReader
 - gdcM::ImageReader, 424
- ImageToImageFilter
 - gdcM::ImageToImageFilter, 426
- ImageWriter
 - gdcM::ImageWriter, 429

- ImplementationClassUIDSub
 - gdcm::network::ImplementationClassUIDSub, 430
- ImplementationUIDSub
 - gdcm::network::ImplementationUIDSub, 431
- ImplementationVersionNameSub
 - gdcm::network::ImplementationVersionNameSub, 431
- Implicit
 - gdcm::TransferSyntax, 710
- ImplicitVRBigEndianACRNEMA
 - gdcm::TransferSyntax, 711
- ImplicitVRBigEndianPrivateGE
 - gdcm::TransferSyntax, 710
- ImplicitVRLittleEndian
 - gdcm::TransferSyntax, 710
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
 - gdcm::UIDs, 724
- IncompleteLUT
 - gdcm::LookupTable, 471
- InitFromRQ
 - gdcm::network::AAssociateACPDU, 137
- Initialize
 - gdcm::network::ULConnectionInfo, 779
- InitializeBlueLUT
 - gdcm::LookupTable, 470
- InitializeConnection
 - gdcm::network::ULConnection, 776
 - gdcm::ServiceClassUser, 630
- InitializeDataSet
 - gdcm::BaseRootQuery, 195
 - gdcm::FindPatientRootQuery, 371
 - gdcm::FindStudyRootQuery, 373
 - gdcm::MovePatientRootQuery, 499
 - gdcm::MoveStudyRootQuery, 501
- InitializeGreenLUT
 - gdcm::LookupTable, 470
- InitializeIncomingConnection
 - gdcm::network::ULConnection, 776
- InitializeLUT
 - gdcm::LookupTable, 470
- InitializeRTStructSet
 - vtkGDCMPolyDataWriter, 828
- InitializeRedLUT
 - gdcm::LookupTable, 470
- Initialized
 - gdcm::LookupTable, 470
- Input
 - gdcm::BitmapToBitmapFilter, 211
- Insert
 - gdcm::CommandDataSet, 238
 - gdcm::DataSet, 282
 - gdcm::FileMetaInformation, 360
 - gdcm::GroupDict, 387
- InsertDataElement
 - gdcm::DataSet, 282
 - gdcm::Item, 446
- InsertEntry
 - gdcm::Table, 691
- InstallPipeline
 - vtkImageColorViewer, 839
- InstanceAvailabilityNotificationSOPClass
 - gdcm::UIDs, 729
- Interactor
 - vtkImageColorViewer, 841
- InteractorStyle
 - vtkImageColorViewer, 842
- Internal
 - gdcm::ApplicationEntity, 155
 - gdcm::Attribute, 168
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
 - gdcm::Element, 320
 - gdcm::Element< VR::AS, VM::VM5 >, 332
 - gdcm::LookupTable, 471
 - gdcm::UI, 717
- InternalCode
 - gdcm::Coder, 231
 - gdcm::JPEG12Codec, 449
 - gdcm::JPEG16Codec, 451
 - gdcm::JPEG8Codec, 456
- InternalReadImageInformation
 - itk::GDCMImageIO2, 381
- Internals
 - vtkRTStructSetProperties, 853
- Invalid
 - gdcm::Usage, 791
- InverseRescale
 - gdcm::Rescaler, 593
- InverseRescaleFunctionIntoBestFit
 - gdcm::Rescaler, 593
- InvokeEvent
 - gdcm::Subject, 670
- IsAETitleValid
 - gdcm::network::AAssociateRQPDU, 142
- IsASCII
 - gdcm::VR, 809
- IsASCII2
 - gdcm::VR, 809
- IsBinary
 - gdcm::VR, 809
- IsBinary2
 - gdcm::VR, 809
- IsDual
 - gdcm::VR, 809
- IsEmpty
 - gdcm::Bitmap, 206
 - gdcm::ByteValue, 217
 - gdcm::CSAElement, 250

- gdcmm::CSAHeaderDict, 257
- gdcmm::Curve, 265
- gdcmm::DataElement, 271
- gdcmm::DataSet, 282
- gdcmm::Defs, 291
- gdcmm::Dict, 298
- gdcmm::Dicts, 307
- gdcmm::Filename, 363
- gdcmm::Macros, 474
- gdcmm::Modules, 497
- gdcmm::Overlay, 514
- gdcmm::Preamble, 553
- gdcmm::PrivateDict, 566
- gdcmm::SegmentHelper::BasicCodedEntry, 198
- IsEncapsulated
 - gdcmm::TransferSyntax, 712
- IsEncoded
 - gdcmm::TransferSyntax, 712
- IsExplicit
 - gdcmm::TransferSyntax, 712
- IsGroupLength
 - gdcmm::Tag, 698
- IsGroupXX
 - gdcmm::Tag, 698
- IsIdentical
 - gdcmm::Filename, 363
- IsIllegal
 - gdcmm::Tag, 698
- IsImage
 - gdcmm::MediaStorage, 482
- IsImplicit
 - gdcmm::TransferSyntax, 712
- IsInPixelData
 - gdcmm::Overlay, 514
- IsKey
 - gdcmm::Scanner, 602
- IsLastFragment
 - gdcmm::network::AAAbortPDU, 134
 - gdcmm::network::AAAssociateACPDU, 137
 - gdcmm::network::AAAssociateRJPDU, 139
 - gdcmm::network::AAAssociateRQPDU, 142
 - gdcmm::network::AReleaseRPPDU, 157
 - gdcmm::network::AReleaseRQPDU, 159
 - gdcmm::network::BasePDU, 192
 - gdcmm::network::PDataTFPDU, 521
- IsLossless
 - gdcmm::PhotometricInterpretation, 532
 - gdcmm::TransferSyntax, 712
- IsLossy
 - gdcmm::Bitmap, 206
 - gdcmm::ImageCodec, 413
 - gdcmm::PhotometricInterpretation, 532
 - gdcmm::TransferSyntax, 712
- IsOdd
 - gdcmm::VL, 800
- IsPrintable
 - gdcmm::ByteValue, 217
- IsPrivate
 - gdcmm::Tag, 698
- IsPrivateCreator
 - gdcmm::Tag, 698
- IsPublic
 - gdcmm::Tag, 699
- IsRetired
 - gdcmm::PhotometricInterpretation, 532
- IsSameColorSpace
 - gdcmm::PhotometricInterpretation, 532
- IsSwap
 - gdcmm::VR, 809
- IsTransferSyntaxCompatible
 - gdcmm::Bitmap, 206
- IsUndefined
 - gdcmm::MediaStorage, 482
 - gdcmm::VL, 800
- IsUndefinedLength
 - gdcmm::DataElement, 271
 - gdcmm::SequenceOfItems, 623
- IsUnique
 - gdcmm::DictEntry, 303
- IsVRFile
 - gdcmm::VR, 809
- IsValid
 - gdcmm::ApplicationEntity, 155
 - gdcmm::CodeString, 234
 - gdcmm::DirectionCosines, 311
 - gdcmm::FileMetaInformation, 360
 - gdcmm::ImageCodec, 413
 - gdcmm::JPEGCodec, 459
 - gdcmm::LO, 467
 - gdcmm::PixelFormat, 536
 - gdcmm::Preamble, 553
 - gdcmm::String, 665
 - gdcmm::TagPath, 702
 - gdcmm::TransferSyntax, 712
 - gdcmm::UIDGenerator, 719
 - gdcmm::VM, 805
 - gdcmm::VR, 809
- IsZero
 - gdcmm::Overlay, 514
- ItFileSetHt
 - gdcmm::SerieHelper, 627
- Item
 - gdcmm::Item, 445
- ItemVector
 - gdcmm::SequenceOfItems, 622
- Items
 - gdcmm::SequenceOfItems, 624
- Iterator

- gdcm::CSAHeaderDict, 256
- gdcm::DataSet, 280
- gdcm::Dict, 297
- gdcm::SequenceOfFragments, 617
- gdcm::SequenceOfItems, 622
- iterator
 - gdcm::CodeString, 233
 - gdcm::LO, 466
 - gdcm::String, 664
- itk, 131
- itk::GDCMImageIO2
 - JPEG, 379
 - JPEG2000, 379
- itk::GDCMImageIO2, 376
 - ~GDCMImageIO2, 380
 - CanReadFile, 380
 - CanWriteFile, 380
 - GDCMImageIO2, 380
 - GetBodyPart, 380
 - GetInstitution, 380
 - GetLabelFromTag, 380
 - GetLoadPrivateTagsDefault, 380
 - GetLoadSequencesDefault, 380
 - GetManufacturer, 380
 - GetModality, 380
 - GetModel, 380
 - GetNumberOfSeriesInStudy, 380
 - GetNumberOfStudyRelatedSeries, 380
 - GetPatientAge, 380
 - GetPatientDOB, 380
 - GetPatientID, 380
 - GetPatientName, 380
 - GetPatientSex, 381
 - GetScanOptions, 381
 - GetStudyDate, 381
 - GetStudyDescription, 381
 - GetStudyID, 381
 - GetValueFromTag, 381
 - InternalReadImageInformation, 381
 - itkBooleanMacro, 381
 - itkGetEnumMacro, 381
 - itkGetMacro, 381
 - itkGetStringMacro, 381, 382
 - itkNewMacro, 382
 - itkSetEnumMacro, 382
 - itkSetMacro, 382
 - itkSetStringMacro, 382
 - itkTypeMacro, 382
 - LoadPrivateTagsDefaultOff, 382
 - LoadPrivateTagsDefaultOn, 382
 - LoadSequencesDefaultOff, 382
 - LoadSequencesDefaultOn, 382
 - m_FrameOfReferenceInstanceUID, 383
 - m_KeepOriginalUID, 383
 - m_RescaleIntercept, 383
 - m_RescaleSlope, 383
 - m_SeriesInstanceUID, 383
 - m_StudyInstanceUID, 383
 - m_UIDPrefix, 384
 - OpenGDCMFileForReading, 383
 - OpenGDCMFileForWriting, 383
 - Pointer, 379
 - PrintSelf, 383
 - Read, 383
 - ReadImageInformation, 383
 - Self, 379
 - SetLoadPrivateTagsDefault, 383
 - SetLoadSequencesDefault, 383
 - Superclass, 379
 - TCompressionType, 379
 - Write, 383
 - WriteImageInformation, 383
- itkBooleanMacro
 - itk::GDCMImageIO2, 381
- itkGDCMImageIO2.h, 946
- itkGetEnumMacro
 - itk::GDCMImageIO2, 381
- itkGetMacro
 - itk::GDCMImageIO2, 381
- itkGetStringMacro
 - itk::GDCMImageIO2, 381, 382
- itkNewMacro
 - itk::GDCMImageIO2, 382
- itkSetEnumMacro
 - itk::GDCMImageIO2, 382
- itkSetMacro
 - itk::GDCMImageIO2, 382
- itkSetStringMacro
 - itk::GDCMImageIO2, 382
- itkTypeMacro
 - itk::GDCMImageIO2, 382
- JPEG
 - itk::GDCMImageIO2, 379
- JPEG2000
 - gdcm::TransferSyntax, 711
 - itk::GDCMImageIO2, 379
- JPEG2000_COMPRESSION
 - vtkGDCMImageWriter, 820
- JPEG2000ImageCompression
 - gdcm::UIDs, 725
- JPEG2000ImageCompressionLosslessOnly
 - gdcm::UIDs, 725
- JPEG2000Lossless
 - gdcm::TransferSyntax, 711
- JPEG2000Part2MulticomponentImageCompression
 - gdcm::UIDs, 725

- JPEG2000Part2MulticomponentImageCompression-
LosslessOnly
 - gdcm::UIDs, 725
- JPEG_COMPRESSION
 - vtkGDCMImageWriter, 820
- JPEGBaselineProcess1
 - gdcm::TransferSyntax, 710
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJP-
EG8BitImageCompression
 - gdcm::UIDs, 724
- JPEGExtendedHierarchicalProcess1618Retired
 - gdcm::UIDs, 725
- JPEGExtendedHierarchicalProcess1719Retired
 - gdcm::UIDs, 725
- JPEGExtendedProcess24DefaultTransferSyntaxforLossy-
JPEG12BitImageCompressionProcess4only
 - gdcm::UIDs, 724
- JPEGExtendedProcess2_4
 - gdcm::TransferSyntax, 710
- JPEGExtendedProcess35Retired
 - gdcm::UIDs, 724
- JPEGExtendedProcess3_5
 - gdcm::TransferSyntax, 710
- JPEGFullProgressionHierarchicalProcess2426Retired
 - gdcm::UIDs, 725
- JPEGFullProgressionHierarchicalProcess2527Retired
 - gdcm::UIDs, 725
- JPEGFullProgressionNonHierarchicalProcess1012-
Retired
 - gdcm::UIDs, 724
- JPEGFullProgressionNonHierarchicalProcess1113-
Retired
 - gdcm::UIDs, 724
- JPEGFullProgressionProcess10_12
 - gdcm::TransferSyntax, 710
- JPEGLS_COMPRESSION
 - vtkGDCMImageWriter, 820
- JPEGLSLossless
 - gdcm::TransferSyntax, 711
- JPEGLSLosslessImageCompression
 - gdcm::UIDs, 725
- JPEGLSLossyNearLosslessImageCompression
 - gdcm::UIDs, 725
- JPEGLSNearLossless
 - gdcm::TransferSyntax, 711
- JPEGLosslessHierarchicalProcess28Retired
 - gdcm::UIDs, 725
- JPEGLosslessHierarchicalProcess29Retired
 - gdcm::UIDs, 725
- JPEGLosslessNonHierarchicalFirstOrderPrediction-
Process14SelectionValue1DefaultTransfer-
SyntaxforLosslessJPEGImageCompression
 - gdcm::UIDs, 725
- JPEGLosslessNonHierarchicalProcess14
 - gdcm::UIDs, 724
- JPEGLosslessNonHierarchicalProcess15Retired
 - gdcm::UIDs, 725
- JPEGLosslessProcess14
 - gdcm::TransferSyntax, 711
- JPEGLosslessProcess14_1
 - gdcm::TransferSyntax, 711
- JPEGSpectralSelectionHierarchicalProcess2022Retired
 - gdcm::UIDs, 725
- JPEGSpectralSelectionHierarchicalProcess2123Retired
 - gdcm::UIDs, 725
- JPEGSpectralSelectionNonHierarchicalProcess68Retired
 - gdcm::UIDs, 724
- JPEGSpectralSelectionNonHierarchicalProcess79Retired
 - gdcm::UIDs, 724
- JPEGSpectralSelectionProcess6_8
 - gdcm::TransferSyntax, 710
- JPIPReferenced
 - gdcm::TransferSyntax, 711
 - gdcm::UIDs, 725
- JPIPReferencedDeflate
 - gdcm::UIDs, 725
- JPEG12Codec
 - gdcm::JPEG12Codec, 449
- JPEG16Codec
 - gdcm::JPEG16Codec, 451
- JPEG2000Codec
 - gdcm::JPEG2000Codec, 453
- JPEG8Codec
 - gdcm::JPEG8Codec, 456
- JPEGCodec
 - gdcm::JPEGCodec, 458
- JPEGLSCodec
 - gdcm::JPEGLSCodec, 462
- Join
 - gdcm::Filename, 363
- JunkAfterDocElementError
 - gdcm::Parser, 518
- KAKADUCodec
 - gdcm::KAKADUCodec, 465
- KeyObjectSelectionDocument
 - gdcm::MediaStorage, 480
- KeyObjectSelectionDocumentStorage
 - gdcm::UIDs, 729
- KeyField
 - gdcm::CSAElement, 251
- KeyValuePairArrayType
 - gdcm::CompositeNetworkFunctions, 241
- KeyValuePairType
 - gdcm::CompositeNetworkFunctions, 241
- LD_ALL
 - gdcm, 119
- LD_NOSEQ

- gdcM, 119
- LD_NOSHADOW
 - gdcM, 119
- LD_NOSHADOWSEQ
 - gdcM, 119
- LINE
 - gdcM::MeshPrimitive, 489
- LO
 - gdcM::VR, 808
- LT
 - gdcM::VR, 808
- LO
 - gdcM::LO, 467
- LOComp
 - gdcM, 117
- LTComp
 - gdcM, 117
- LUT
 - gdcM::Bitmap, 208
 - gdcM::ImageCodec, 415
- LUTPtr
 - gdcM::Bitmap, 204
 - gdcM::ImageCodec, 412
- LeadECGWaveformStorage
 - gdcM::MediaStorage, 480
- Level
 - vtkImageMapToWindowLevelColors2, 846
- ListCharSets
 - gdcM::QueryFactory, 576
- LittleEndian
 - gdcM::SwapCode, 684
- Load
 - gdcM::Directory, 313
- LoadDefault
 - gdcM::CSAHeaderDict, 257
 - gdcM::Dict, 298
 - gdcM::PrivateDict, 566
- LoadDefaults
 - gdcM::Defs, 291
 - gdcM::Dicts, 307
- LoadFromDataElement
 - gdcM::CSAHeader, 255
 - gdcM::PDBHeader, 525
- LoadFromFile
 - gdcM::Defs, 291
- LoadIconImage
 - vtkGDCMImageReader, 818
- LoadImageFromFiles
 - gdcM::DirectoryHelper, 315
- LoadOverlays
 - vtkGDCMImageReader, 818
- LoadPrivateTagsDefaultOff
 - itk::GDCMImageIO2, 382
- LoadPrivateTagsDefaultOn
 - itk::GDCMImageIO2, 382
- LoadResourcesFiles
 - gdcM::Global, 385
- LoadSequencesDefaultOff
 - itk::GDCMImageIO2, 382
- LoadSequencesDefaultOn
 - itk::GDCMImageIO2, 382
- LoadSingleFile
 - vtkGDCMImageReader, 816
- Locate
 - gdcM::Global, 386
- LodModeType
 - gdcM, 119
- LookupTable
 - gdcM::LookupTable, 469
 - vtkImageMapToColors16, 844
- LookupTableType
 - gdcM::LookupTable, 469
- Lossless
 - gdcM::JPEGCodec, 460
- LossyFlag
 - gdcM::Bitmap, 208
 - gdcM::ImageCodec, 415
 - vtkGDCMImageReader, 818
- MAGNIFIED
 - gdcM::Spacing, 648
- MANUAL
 - gdcM::Segment, 606
- MONOCHROME1
 - gdcM::PhotometricInterpretation, 532
- MONOCHROME2
 - gdcM::PhotometricInterpretation, 532
- MPEG2MainProfile
 - gdcM::TransferSyntax, 711
- MPEG2MainProfileMainLevel
 - gdcM::UIDs, 725
- MPTType_END
 - gdcM::MeshPrimitive, 489
- MRImageStorage
 - gdcM::MediaStorage, 479
 - gdcM::UIDs, 727
- MRSpectroscopyStorage
 - gdcM::MediaStorage, 479
 - gdcM::UIDs, 727
- MS_END
 - gdcM::MediaStorage, 481
- m_ConstMemberFunction
 - gdcM::MemberCommand, 486
- m_FrameOfReferenceInstanceUID
 - itk::GDCMImageIO2, 383
- m_KeepOriginalUID
 - itk::GDCMImageIO2, 383
- m_MemberFunction

- gdcM::MemberCommand, 486
- gdcM::SimpleMemberCommand, 637
- m_RescaleIntercept
 - itk::GDCMImageIO2, 383
- m_RescaleSlope
 - itk::GDCMImageIO2, 383
- m_SeriesInstanceUID
 - itk::GDCMImageIO2, 383
- m_StudyInstanceUID
 - itk::GDCMImageIO2, 383
- m_This
 - gdcM::MemberCommand, 486
 - gdcM::SimpleMemberCommand, 637
- m_UIDPrefix
 - itk::GDCMImageIO2, 384
- m_char
 - gdcM::ignore_char, 393
- mAction
 - gdcM::network::Transition, 715
- MD5
 - gdcM::MD5, 476
- MD5DataImagesType
 - gdcM::Testing, 704
- MD5MetalImagesType
 - vtkGDCMTesting, 830
- mDataSet
 - gdcM::BaseRootQuery, 196
- mElementOffsets
 - gdcM::StreamImageWriter, 661
- mElementOffsets1
 - gdcM::StreamImageWriter, 661
- mEnd
 - gdcM::network::Transition, 715
- mFileOffset
 - gdcM::StreamImageReader, 656
- mFileOffset1
 - gdcM::StreamImageReader, 656
- mHeaderInformation
 - gdcM::StreamImageReader, 656
- mHelpDescription
 - gdcM::BaseRootQuery, 196
- mImage
 - gdcM::BaseRootQuery, 196
- MPType
 - gdcM::MeshPrimitive, 489
- mPatient
 - gdcM::BaseRootQuery, 196
- mReader
 - gdcM::StreamImageReader, 656
- mRootType
 - gdcM::BaseRootQuery, 196
- MSType
 - gdcM::MediaStorage, 479
- mSeries
 - gdcM::BaseRootQuery, 196
- mStudy
 - gdcM::BaseRootQuery, 196
- mWriter
 - gdcM::StreamImageWriter, 662
- mXMax
 - gdcM::StreamImageReader, 656
 - gdcM::StreamImageWriter, 662
- mXMin
 - gdcM::StreamImageReader, 656
 - gdcM::StreamImageWriter, 662
- mYMax
 - gdcM::StreamImageReader, 656
 - gdcM::StreamImageWriter, 662
- mYMin
 - gdcM::StreamImageReader, 656
 - gdcM::StreamImageWriter, 662
- mZMax
 - gdcM::StreamImageReader, 656
 - gdcM::StreamImageWriter, 662
- mZMin
 - gdcM::StreamImageReader, 657
 - gdcM::StreamImageWriter, 662
- Macro
 - gdcM::Macro, 472
- MacroEntry
 - gdcM, 117
- Macros
 - gdcM::Macros, 474
- magenta
 - gdcM::terminal, 130
- MakeDirectory
 - gdcM::System, 689
- MakeNew
 - gdcM::network::Transition, 715
- MakeObject
 - gdcM::AnonymizeEvent, 146
 - gdcM::DataEvent, 277
 - gdcM::DataSetEvent, 286
 - gdcM::Event, 341
 - gdcM::ProgressEvent, 570
- MammographyCADSR
 - gdcM::MediaStorage, 480
- MammographyCADSRStorage
 - gdcM::UIDs, 728
- Mandatory
 - gdcM::Usage, 791
- MapCSAHeaderDictEntry
 - gdcM::CSAHeaderDict, 256
- MapDictEntry
 - gdcM::Dict, 297
- MapIODEntry
 - gdcM::IOD, 435
- MapModuleEntry

- gdcmmacro::Macro, 472
- gdcmmodule::Module, 492
- MapScalarsThroughTable2
 - vtkLookupTable16, 849
- MapTableEntry
 - gdcmmodule::Table, 691
- MappingType
 - gdcmmodule::Scanner, 600
- MaxLength
 - gdcmmodule::ApplicationEntity, 155
 - gdcmmodule::PersonName, 530
- MaxNumberOfComponents
 - gdcmmodule::ApplicationEntity, 155
 - gdcmmodule::PersonName, 530
- MaxPrintLength
 - gdcmmodule::Printer, 565
- MaximumLengthSub
 - gdcmmodule::network::MaximumLengthSub, 475
- MediaCreationManagementSOPClassUID
 - gdcmmodule::UIDs, 727
- MediaStorageDirectoryStorage
 - gdcmmodule::MediaStorage, 479
 - gdcmmodule::UIDs, 725
- MediaStorage
 - gdcmmodule::MediaStorage, 481
- MediaStorageDataFilesType
 - gdcmmodule::Testing, 704
- MedicalImageProperties
 - vtkGDCMImageReader, 818
 - vtkGDCMPolyDataReader, 826
 - vtkGDCMPolyDataWriter, 829
- MemberCommand
 - gdcmmodule::MemberCommand, 485
- MeshPrimitive
 - gdcmmodule::MeshPrimitive, 489
- MessageID
 - gdcmmodule::network::CEchoRQ, 220
- MetaInformationTS
 - gdcmmodule::FileMetaInformation, 362
- ModalityPerformedProcedureStepNotificationSOPClass
 - gdcmmodule::UIDs, 726
- ModalityPerformedProcedureStepRetrieveSOPClass
 - gdcmmodule::UIDs, 726
- ModalityPerformedProcedureStepSOPClass
 - gdcmmodule::MediaStorage, 481
 - gdcmmodule::UIDs, 726
- ModalityWorklistInformationModelFIND
 - gdcmmodule::UIDs, 729
- Mode
 - gdcmmodule::terminal, 130
- Module
 - gdcmmodule::Module, 492
- ModuleEntry
 - gdcmmodule::ModuleEntry, 495
- ModuleMapType
 - gdcmmodule::Macros, 474
 - gdcmmodule::Modules, 497
- Modules
 - gdcmmodule::Modules, 497
- MovePatientRootQuery
 - gdcmmodule::MovePatientRootQuery, 499
- MoveStudyRootQuery
 - gdcmmodule::MoveStudyRootQuery, 501
- mshFile
 - gdcmmodule::StreamImageWriter, 662
- MultiframeGrayscaleByteSecondaryCaptureImageStorage
 - gdcmmodule::MediaStorage, 479
 - gdcmmodule::UIDs, 727
- MultiframeGrayscaleWordSecondaryCaptureImageStorage
 - gdcmmodule::MediaStorage, 479
 - gdcmmodule::UIDs, 727
- MultiframeSingleBitSecondaryCaptureImageStorage
 - gdcmmodule::MediaStorage, 479
 - gdcmmodule::UIDs, 727
- MultiframeTrueColorSecondaryCaptureImageStorage
 - gdcmmodule::MediaStorage, 480
 - gdcmmodule::UIDs, 727
- N_ACTION_RQ
 - gdcmmodule::network::DIMSE, 309
- N_ACTION_RSP
 - gdcmmodule::network::DIMSE, 309
- N_CREATE_RQ
 - gdcmmodule::network::DIMSE, 309
- N_CREATE_RSP
 - gdcmmodule::network::DIMSE, 309
- N_DELETE_RQ
 - gdcmmodule::network::DIMSE, 309
- N_DELETE_RSP
 - gdcmmodule::network::DIMSE, 309
- N_EVENT_REPORT_RQ
 - gdcmmodule::network::DIMSE, 309
- N_EVENT_REPORT_RSP
 - gdcmmodule::network::DIMSE, 309
- N_GET_RQ
 - gdcmmodule::network::DIMSE, 309
- N_GET_RSP
 - gdcmmodule::network::DIMSE, 309
- N_SET_RQ
 - gdcmmodule::network::DIMSE, 309
- N_SET_RSP
 - gdcmmodule::network::DIMSE, 309
- NO
 - gdcmmodule::Surface, 673
- NO_COMPRESSION
 - vtkGDCMImageWriter, 820

- NOMAGIC
 - gdcm::CSAHeader, 253
- Name
 - gdcm::ModuleEntry, 496
- NameField
 - gdcm::CSAElement, 251
 - gdcm::PDBelement, 524
- NeedByteSwap
 - gdcm::Bitmap, 208
 - gdcm::ImageCodec, 415
- NeedOverlayCleanup
 - gdcm::ImageCodec, 415
- NegotiatedType
 - gdcm::TransferSyntax, 710
- NestedMacroEntries
 - gdcm, 117
- NestedModuleEntries
 - gdcm::NestedModuleEntries, 504
- New
 - gdcm::Anonymizer, 150
 - gdcm::MemberCommand, 486
 - gdcm::Scanner, 603
 - gdcm::SequenceOfFragments, 618
 - gdcm::SequenceOfItems, 623
 - gdcm::SimpleMemberCommand, 637
 - vtkGDCMImageReader, 816
 - vtkGDCMImageWriter, 821
 - vtkGDCMMedicalImageProperties, 823
 - vtkGDCMPolyDataReader, 825
 - vtkGDCMPolyDataWriter, 828
 - vtkGDCMTesting, 830
 - vtkGDCMThreadedImageReader, 832
 - vtkGDCMThreadedImageReader2, 834
 - vtkImageColorViewer, 839
 - vtkImageMapToColors16, 843
 - vtkImageMapToWindowLevelColors2, 845
 - vtkImagePlanarComponentsToComponents, 846
 - vtkImageRGBToYBR, 847
 - vtkImageYBRToRGB, 848
 - vtkLookupTable16, 849
 - vtkRTStructSetProperties, 852
- NoElementsError
 - gdcm::Parser, 518
- NoError
 - gdcm::Parser, 518
- NoMemoryError
 - gdcm::Parser, 518
- NoObject
 - gdcm::MediaStorage, 481
- NoOfItemsField
 - gdcm::CSAElement, 251
- Normalize
 - gdcm::DirectionCosines, 311
- NuclearMedicineImageStorage
 - gdcm::MediaStorage, 480
 - gdcm::UIDs, 728
- NuclearMedicineImageStorageRetired
 - gdcm::MediaStorage, 479
 - gdcm::UIDs, 727
- NumberOfDimensions
 - gdcm::Bitmap, 208
 - gdcm::ImageCodec, 415
- NumberOfIconImages
 - vtkGDCMImageReader, 818
- NumberOfOverlays
 - vtkGDCMImageReader, 818
- NumberOfSurfaces
 - gdcm::SurfaceWriter, 683
- OB
 - gdcm::VR, 808
- OB_OW
 - gdcm::VR, 808
- OBLIQUE
 - gdcm::Orientation, 509
- OF
 - gdcm::VR, 808
- OW
 - gdcm::VR, 808
- Object
 - gdcm::Object, 507
- ObjectEnd
 - gdcm::MediaStorage, 481
- ObjectType
 - gdcm::MediaStorage, 481
- Ofstream
 - gdcm::Writer, 859
- OneShotReadBuf
 - gdcm::OneShotReadBuf, 508
- op
 - gdcm::SerieHelper::Rule, 597
- OpenGDCMFileForReading
 - itk::GDCMImageIO2, 383
- OpenGDCMFileForWriting
 - itk::GDCMImageIO2, 383
- operator const char *
 - gdcm::ConstCharWrapper, 243
 - gdcm::Filename, 363
 - gdcm::String, 665
- operator const double *
 - gdcm::DirectionCosines, 311
- operator const std::vector< char > &
 - gdcm::ByteValue, 217
- operator MStype
 - gdcm::MediaStorage, 482
- operator ObjectType *
 - gdcm::SmartPointer, 641
- operator PType

- gdcmm::PhotometricInterpretation, 533
- operator ScalarType
 - gdcmm::PixelFormat, 536
- operator SwapCode::SwapCodeType
 - gdcmm::SwapCode, 685
- operator TSType
 - gdcmm::TransferSyntax, 712
 - gdcmm::UIDs, 738
- operator TypeType
 - gdcmm::Type, 716
- operator uint32_t
 - gdcmm::VL, 800
- operator UsageType
 - gdcmm::Usage, 792
- operator VMType
 - gdcmm::VM, 805
- operator VRType
 - gdcmm::VR, 809
- operator<
 - gdcmm::Attribute, 166
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcmm::CSAElement, 250
 - gdcmm::CSAHeaderDictEntry, 258
 - gdcmm::DataElement, 271
 - gdcmm::PrivateTag, 568
 - gdcmm::Tag, 699
- operator<<
 - gdcmm, 119d
 - gdcmm::BasicOffsetTable, 201
 - gdcmm::CodeString, 234
 - gdcmm::CommandDataSet, 238
 - gdcmm::CSAElement, 251
 - gdcmm::CSAHeader, 255
 - gdcmm::CSAHeaderDict, 257
 - gdcmm::CSAHeaderDictEntry, 259
 - gdcmm::DataElement, 274
 - gdcmm::DataSet, 284
 - gdcmm::Dict, 298
 - gdcmm::DictEntry, 303
 - gdcmm::Dicts, 307
 - gdcmm::Directory, 314
 - gdcmm::File, 352
 - gdcmm::FileMetaInformation, 362
 - gdcmm::FileSet, 367
 - gdcmm::Fragment, 376
 - gdcmm::Global, 386
 - gdcmm::GroupDict, 388
 - gdcmm::IOD, 436
 - gdcmm::IODEntry, 438
 - gdcmm::IODs, 439
 - gdcmm::Item, 446
 - gdcmm::Macro, 473
 - gdcmm::Macros, 474
- gdcmm::MediaStorage, 483
- gdcmm::Module, 493
- gdcmm::ModuleEntry, 495
- gdcmm::Modules, 497
- gdcmm::NestedModuleEntries, 504
- gdcmm::Object, 507
- gdcmm::Orientation, 510
- gdcmm::PDSElement, 524
- gdcmm::PDBHeader, 526
- gdcmm::PhotometricInterpretation, 533
- gdcmm::PixelFormat, 537
- gdcmm::Preamble, 553
- gdcmm::PrivateDict, 566
- gdcmm::PrivateTag, 568
- gdcmm::Scanner, 603
- gdcmm::Sorter, 646
- gdcmm::SwapCode, 685
- gdcmm::Table, 691
- gdcmm::Tag, 701
- gdcmm::TransferSyntax, 712
- gdcmm::Type, 717
- gdcmm::UI, 717
- gdcmm::Usage, 792
- gdcmm::Version, 799
- gdcmm::VL, 801
- gdcmm::VM, 805
- gdcmm::VR, 810
- operator<=
 - gdcmm::Tag, 699
- operator>>
 - gdcmm, 123
 - gdcmm::Tag, 701
- operator*
 - gdcmm::SmartPointer, 641
- operator()
 - gdcmm::DataSet, 283
 - gdcmm::Scanner::Itstr, 471
- operator++
 - gdcmm::VL, 801
- operator+=
 - gdcmm::VL, 801
- operator->
 - gdcmm::SmartPointer, 641
- operator=
 - gdcmm::ByteValue, 217
 - gdcmm::CSAElement, 250
 - gdcmm::DataElement, 272
 - gdcmm::DataSet, 283
 - gdcmm::Element< TVR, VM::VM1_n >, 324
 - gdcmm::Object, 507
 - gdcmm::ParseException, 517
 - gdcmm::Preamble, 553
 - gdcmm::SequenceOfItems, 623
 - gdcmm::SmartPointer, 641

- gdcmm::Tag, 699
- operator==
 - gdcmm, 123
 - gdcmm::Attribute, 166
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcmm::ByteValue, 217
 - gdcmm::CodeString, 234
 - gdcmm::CSAElement, 250
 - gdcmm::DataElement, 272
 - gdcmm::network::AbstractSyntax, 144
 - gdcmm::network::PresentationContextRQ, 560
 - gdcmm::network::TransferSyntaxSub, 713
 - gdcmm::PDBelement, 523
 - gdcmm::PixelFormat, 537
 - gdcmm::PresentationContext, 554
 - gdcmm::SequenceOfFragments, 618
 - gdcmm::SequenceOfItems, 623
 - gdcmm::Tag, 699
 - gdcmm::Value, 797
- OphthalmicPhotography16BitImageStorage
 - gdcmm::UIDs, 728
- OphthalmicPhotography8BitImageStorage
 - gdcmm::UIDs, 728
- OphthalmicTomographyImageStorage
 - gdcmm::UIDs, 728
- OrderFileList
 - gdcmm::SerieHelper, 627
- Orientation
 - gdcmm::Orientation, 509
- OrientationType
 - gdcmm::Orientation, 509
- Output
 - gdcmm::BitmapToBitmapFilter, 211
- OutputFormat
 - vtkImageMapToColors16, 844
- OutputTypes
 - gdcmm::DictConverter, 300
- Overlay
 - gdcmm::Overlay, 513
- OverlayImageActor
 - vtkImageColorViewer, 842
- Overlays
 - gdcmm::Pixmap, 541
- PALETTE_COLOR
 - gdcmm::PhotometricInterpretation, 532
- PDF
 - gdcmm::MediaStorage, 481
- PETImageStorage
 - gdcmm::MediaStorage, 480
- PHILIPS
 - gdcmm::Dicts, 306
- PI_END
 - gdcmm::PhotometricInterpretation, 532
- PN
 - gdcmm::VR, 808
- POINTS
 - gdcmm::Surface, 674
- PDBelement
 - gdcmm::PDBelement, 523
- PDBHeader
 - gdcmm::PDBHeader, 525
- PDFCodec
 - gdcmm::PDFCodec, 527
- PDataTFPDU
 - gdcmm::network::PDataTFPDU, 521
- PF
 - gdcmm::Bitmap, 208
 - gdcmm::ImageCodec, 415
- PI
 - gdcmm::Bitmap, 208
 - gdcmm::ImageCodec, 415
- PIType
 - gdcmm::PhotometricInterpretation, 532
- PNComp
 - gdcmm, 117
- PNMCodec
 - gdcmm::PNMCodec, 551
- PVRGCodec
 - gdcmm::PVRGCodec, 572
- Pack
 - gdcmm::Unpacker12Bits, 790
- Padding
 - gdcmm::ApplicationEntity, 155
 - gdcmm::PersonName, 530
- Parent
 - gdcmm::Element< TVR, VM::VM1_2 >, 322
 - gdcmm::Element< TVR, VM::VM2_2n >, 327
 - gdcmm::Element< TVR, VM::VM2_n >, 329
 - gdcmm::Element< TVR, VM::VM3_3n >, 330
 - gdcmm::Element< TVR, VM::VM3_n >, 332
- Parse
 - gdcmm::Parser, 519
- ParseBuffer
 - gdcmm::Parser, 519
- ParseCertificateFile
 - gdcmm::CryptographicMessageSyntax, 247
- ParseDateTime
 - gdcmm::System, 689, 690
- ParseDump
 - gdcmm::ASN1, 161
- ParseDumpFile
 - gdcmm::ASN1, 161
- ParseException
 - gdcmm::ParseException, 517
- ParseKeyFile
 - gdcmm::CryptographicMessageSyntax, 247

- Parser
 - gdcm::Parser, 519
- PassAlphaToOutput
 - vtkImageMapToColors16, 844
- Patient
 - gdcm::Patient, 520
- PatientRootQueryRetrieveInformationModelFIND
 - gdcm::UIDs, 729
- PatientRootQueryRetrieveInformationModelGET
 - gdcm::UIDs, 729
- PatientRootQueryRetrieveInformationModelMOVE
 - gdcm::UIDs, 729
- PatientStudyOnlyQueryRetrieveInformationModelFIND-Retired
 - gdcm::UIDs, 729
- PatientStudyOnlyQueryRetrieveInformationModelGET-Retired
 - gdcm::UIDs, 729
- PatientStudyOnlyQueryRetrieveInformationModelMOVE-Retired
 - gdcm::UIDs, 729
- PerformAction
 - gdcm::network::ULAction, 740
 - gdcm::network::ULActionAA1, 741
 - gdcm::network::ULActionAA2, 742
 - gdcm::network::ULActionAA3, 744
 - gdcm::network::ULActionAA4, 745
 - gdcm::network::ULActionAA5, 746
 - gdcm::network::ULActionAA6, 747
 - gdcm::network::ULActionAA7, 748
 - gdcm::network::ULActionAA8, 749
 - gdcm::network::ULActionAE1, 751
 - gdcm::network::ULActionAE2, 752
 - gdcm::network::ULActionAE3, 753
 - gdcm::network::ULActionAE4, 754
 - gdcm::network::ULActionAE5, 755
 - gdcm::network::ULActionAE6, 756
 - gdcm::network::ULActionAE7, 758
 - gdcm::network::ULActionAE8, 759
 - gdcm::network::ULActionAR1, 760
 - gdcm::network::ULActionAR10, 761
 - gdcm::network::ULActionAR2, 762
 - gdcm::network::ULActionAR3, 763
 - gdcm::network::ULActionAR4, 765
 - gdcm::network::ULActionAR5, 766
 - gdcm::network::ULActionAR6, 767
 - gdcm::network::ULActionAR7, 768
 - gdcm::network::ULActionAR8, 769
 - gdcm::network::ULActionAR9, 770
 - gdcm::network::ULActionDT1, 772
 - gdcm::network::ULActionDT2, 773
- Philips3D
 - gdcm::MediaStorage, 480
- PhilipsPrivateMRSyntheticImageStorage
 - gdcm::MediaStorage, 481
- PhotometricInterpretation
 - gdcm::PhotometricInterpretation, 532
- PixelData
 - gdcm::Bitmap, 208
 - gdcm::PixmapReader, 544
 - gdcm::PixmapWriter, 549
- PixelFormat
 - gdcm::PixelFormat, 535
- Pixmap
 - gdcm::Pixmap, 540
- PixmapReader
 - gdcm::Bitmap, 208
 - gdcm::PixmapReader, 543
- PixmapToPixmapFilter
 - gdcm::PixmapToPixmapFilter, 545
- PixmapWriter
 - gdcm::PixmapWriter, 548
- PlanarConfiguration
 - gdcm::Bitmap, 208
 - gdcm::ImageCodec, 415
 - vtkGDCMImageReader, 818
- Pointer
 - itk::GDCMImageIO2, 379
- pointer
 - gdcm::CodeString, 233
 - gdcm::LO, 466
 - gdcm::String, 664
- PositronEmissionTomographyImageStorage
 - gdcm::UIDs, 729
- Preamble
 - gdcm::Preamble, 552
- PrepareWrite
 - gdcm::PixmapWriter, 549
 - gdcm::SegmentWriter, 614
 - gdcm::SurfaceWriter, 683
- PrepareWritePointMacro
 - gdcm::SurfaceWriter, 683
- Prepend
 - gdcm::Global, 386
- PresentationLUTSOPClass
 - gdcm::UIDs, 727
- PresentationContext
 - gdcm::PresentationContext, 554
- PresentationContextAC
 - gdcm::network::PresentationContextAC, 555
- PresentationContextArrayType
 - gdcm::network::AAssociateRQPDU, 141
 - gdcm::PresentationContextGenerator, 557
- PresentationContextGenerator
 - gdcm::PresentationContextGenerator, 557
- PresentationContextRQ
 - gdcm::network::PresentationContextRQ, 559
- PresentationDataValue

- gdcmm::network::PresentationDataValue, 561
- PrimitiveData
 - gdcmm::MeshPrimitive, 490
- PrimitiveType
 - gdcmm::MeshPrimitive, 490
- PrimitivesData
 - gdcmm::MeshPrimitive, 489
- Print
 - gdcmm::ApplicationEntity, 155
 - gdcmm::Attribute, 167
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 179
 - gdcmm::Bitmap, 206
 - gdcmm::ByteValue, 217
 - gdcmm::CSAHeader, 255
 - gdcmm::Curve, 265
 - gdcmm::DataSet, 283
 - gdcmm::DictPrinter, 305
 - gdcmm::DirectionCosines, 311
 - gdcmm::Directory, 313
 - gdcmm::Element, 320
 - gdcmm::Element< TVR, VM::VM1_n >, 324
 - gdcmm::Element< VR::AS, VM::VM5 >, 332
 - gdcmm::Event, 341
 - gdcmm::Image, 396
 - gdcmm::LookupTable, 470
 - gdcmm::network::AAAbortPDU, 134
 - gdcmm::network::AAssociateACPDU, 137
 - gdcmm::network::AAssociateRJPDU, 139
 - gdcmm::network::AAssociateRQPDU, 142
 - gdcmm::network::AbstractSyntax, 144
 - gdcmm::network::ApplicationContext, 154
 - gdcmm::network::AReleaseRPPDU, 157
 - gdcmm::network::AReleaseRQPDU, 159
 - gdcmm::network::BasePDU, 192
 - gdcmm::network::PDataTFPDU, 521
 - gdcmm::network::PresentationContextAC, 555
 - gdcmm::network::PresentationContextRQ, 560
 - gdcmm::network::PresentationDataValue, 561
 - gdcmm::network::TransferSyntaxSub, 713
 - gdcmm::network::UserInfo, 794
 - gdcmm::Object, 507
 - gdcmm::Orientation, 509
 - gdcmm::Overlay, 514
 - gdcmm::PDBHeader, 525
 - gdcmm::PersonName, 530
 - gdcmm::PixelFormat, 537
 - gdcmm::Pixmap, 540
 - gdcmm::Preamble, 553
 - gdcmm::PresentationContext, 554
 - gdcmm::Printer, 564
 - gdcmm::Scanner, 603
 - gdcmm::SegmentedPaletteColorLookupTable, 609
 - gdcmm::SequenceOfFragments, 618
 - gdcmm::SequenceOfItems, 623
 - gdcmm::Sorter, 646
 - gdcmm::TagPath, 702
 - gdcmm::Testing, 706
 - gdcmm::Version, 799
- PrintJobSOPClass
 - gdcmm::UIDs, 726
- PrintQueueManagementSOPClassRetired
 - gdcmm::UIDs, 727
- PrintQueueSOPInstanceRetired
 - gdcmm::UIDs, 727
- PrintASCII
 - gdcmm::ByteValue, 217
- PrintAsPipeSeparatedString
 - gdcmm::Tag, 699
- PrintDataElement
 - gdcmm::Printer, 564
- PrintDataElement2
 - gdcmm::DictPrinter, 305
- PrintDataSet
 - gdcmm::Printer, 564
- PrintDataSet2
 - gdcmm::DictPrinter, 305
- PrintGroupLength
 - gdcmm::ByteValue, 218
- PrintHex
 - gdcmm::ByteValue, 218
- PrintSQ
 - gdcmm::Printer, 564
- PrintSelf
 - itk::GDCMImageIO2, 383
 - vtkGDCMImageReader, 816
 - vtkGDCMImageWriter, 821
 - vtkGDCMMedicalImageProperties, 823
 - vtkGDCMPolyDataReader, 825
 - vtkGDCMPolyDataWriter, 828
 - vtkGDCMTesting, 830
 - vtkGDCMThreadedImageReader, 832
 - vtkGDCMThreadedImageReader2, 834
 - vtkImageColorViewer, 839
 - vtkImageMapToColors16, 843
 - vtkImageMapToWindowLevelColors2, 845
 - vtkImagePlanarComponentsToComponents, 846
 - vtkImageRGBToYBR, 847
 - vtkImageYBRToRGB, 848
 - vtkLookupTable16, 849
 - vtkRTStructSetProperties, 852
- PrintStyle
 - gdcmm::Printer, 565
- PrintStyles
 - gdcmm::Printer, 564
- PrintTable

- gdcmm::network::ULTransitionTable, 784
- PrintXML
 - gdcmm::PrivateDict, 566
- Printer
 - gdcmm::Printer, 564
- PrinterConfigurationRetrievalSOPClass
 - gdcmm::UIDs, 726
- PrinterConfigurationRetrievalSOPInstance
 - gdcmm::UIDs, 726
- PrinterSOPClass
 - gdcmm::UIDs, 726
- PrinterSOPInstance
 - gdcmm::UIDs, 726
- PrivateDict
 - gdcmm::PrivateDict, 566
- PrivateTag
 - gdcmm::PrivateTag, 568
- ProceduralEventLoggingSOPClass
 - gdcmm::UIDs, 726
- ProceduralEventLoggingSOPInstance
 - gdcmm::UIDs, 726
- ProcedureLogStorage
 - gdcmm::UIDs, 728
- Process
 - gdcmm::Parser, 519
- ProcessDataSet
 - gdcmm::FileExplicitFilter, 356
- ProcessPublicTag
 - gdcmm::Scanner, 603
- ProduceCharacterSetDataElement
 - gdcmm::QueryFactory, 576
- ProduceQuery
 - gdcmm::QueryFactory, 576
- ProductCharacteristicsQuerySOPClass
 - gdcmm::UIDs, 730
- ProgressEvent
 - gdcmm::ProgressEvent, 570
- PropertyCategory
 - gdcmm::Segment, 607
- PropertyType
 - gdcmm::Segment, 607
- PseudoColorSoftcopyPresentationStateStorageSOP-
Class
 - gdcmm::UIDs, 728
- PullPrintRequestSOPClassRetired
 - gdcmm::UIDs, 727
- PullStoredPrintManagementMetaSOPClassRetired
 - gdcmm::UIDs, 727
- Push
 - gdcmm::TagPath, 702
- PushBackFile
 - vtkGDCMMedicalImageProperties, 823
- PythonFilter
 - gdcmm::PythonFilter, 573
- Quality
 - gdcmm::JPEGCodec, 460
- QueryFactory
 - gdcmm::BaseRootQuery, 196
 - gdcmm::FindPatientRootQuery, 371
 - gdcmm::FindStudyRootQuery, 373
 - gdcmm::MovePatientRootQuery, 499
 - gdcmm::MoveStudyRootQuery, 502
- RED
 - gdcmm::LookupTable, 469
- RFC2557MIMEencapsulation
 - gdcmm::UIDs, 725
- RGB
 - gdcmm::PhotometricInterpretation, 532
- RLE_COMPRESSION
 - vtkGDCMImageWriter, 820
- RLELossless
 - gdcmm::TransferSyntax, 711
 - gdcmm::UIDs, 725
- RTBeamsDeliveryInstructionStorageSupplement74-
FrozenDraft
 - gdcmm::UIDs, 729
- RTBeamsTreatmentRecordStorage
 - gdcmm::UIDs, 729
- RTBrachyTreatmentRecordStorage
 - gdcmm::UIDs, 729
- RTConventionalMachineVerificationSupplement74Frozen-
Draft
 - gdcmm::UIDs, 729
- RTDoseStorage
 - gdcmm::MediaStorage, 480
 - gdcmm::UIDs, 729
- RTImageStorage
 - gdcmm::MediaStorage, 480
 - gdcmm::UIDs, 729
- RTIonBeamsTreatmentRecordStorage
 - gdcmm::MediaStorage, 481
 - gdcmm::UIDs, 729
- RTIonMachineVerificationSupplement74FrozenDraft
 - gdcmm::UIDs, 729
- RTIonPlanStorage
 - gdcmm::MediaStorage, 481
 - gdcmm::UIDs, 729
- RTPlanStorage
 - gdcmm::MediaStorage, 480
 - gdcmm::UIDs, 729
- RTStructureSetStorage
 - gdcmm::MediaStorage, 480
 - gdcmm::UIDs, 729
- RTTreatmentSummaryRecordStorage
 - gdcmm::UIDs, 729
- RAWCodec
 - gdcmm::RAWCodec, 585

- README.txt, 947
- RGB2YBR
 - gdcm::ImageChangePhotometricInterpretation, 402
- RGBPixelsToRGBPlanes
 - gdcm::ImageChangePlanarConfiguration, 405
- RGBPlanesToRGBPixels
 - gdcm::ImageChangePlanarConfiguration, 405
- RGBToRecommendedDisplayCIELab
 - gdcm::SurfaceHelper, 678
- RGBToRecommendedDisplayGrayscale
 - gdcm::SurfaceHelper, 679
- RLECodec
 - gdcm::RLECodec, 595
- RTStructSetProperties
 - vtkGDCMPolyDataReader, 826
 - vtkGDCMPolyDataWriter, 829
- RawDataStorage
 - gdcm::MediaStorage, 480
 - gdcm::UIDs, 728
- Read
 - gdcm::BasicOffsetTable, 200
 - gdcm::ByteValue, 218
 - gdcm::CommandDataSet, 238
 - gdcm::CP246ExplicitDataElement, 245
 - gdcm::CSAHeader, 255
 - gdcm::DataElement, 272
 - gdcm::DataSet, 283
 - gdcm::Element, 320
 - gdcm::Element< TVR, VM::VM1_n >, 324
 - gdcm::EncodingImplementation< VR::VRASCII >, 337
 - gdcm::EncodingImplementation< VR::VRBINARY >, 338
 - gdcm::ExplicitDataElement, 346
 - gdcm::ExplicitImplicitDataElement, 348
 - gdcm::File, 352
 - gdcm::FileMetaInformation, 361
 - gdcm::Fragment, 376
 - gdcm::ImageReader, 424
 - gdcm::ImplicitDataElement, 433
 - gdcm::Item, 446
 - gdcm::network::AAAbortPDU, 134
 - gdcm::network::AAAssociateACPDU, 137
 - gdcm::network::AAAssociateRJPDU, 139
 - gdcm::network::AAAssociateRQPDU, 142
 - gdcm::network::AbstractSyntax, 144
 - gdcm::network::ApplicationContext, 154
 - gdcm::network::AReleaseRPPDU, 157
 - gdcm::network::AReleaseRQPDU, 159
 - gdcm::network::AsynchronousOperationsWindow-Sub, 162
 - gdcm::network::BasePDU, 192
 - gdcm::network::ImplementationClassUIDSub, 430
 - gdcm::network::ImplementationVersionNameSub, 431
 - gdcm::network::MaximumLengthSub, 475
 - gdcm::network::PDataTFPDU, 521
 - gdcm::network::PresentationContextAC, 555
 - gdcm::network::PresentationContextRQ, 560
 - gdcm::network::PresentationDataValue, 561
 - gdcm::network::TransferSyntaxSub, 713
 - gdcm::network::UserInformation, 794
 - gdcm::PixmapReader, 543
 - gdcm::PNMCodec, 551
 - gdcm::Preamble, 553
 - gdcm::Reader, 589
 - gdcm::SegmentReader, 612
 - gdcm::SequenceOfFragments, 618
 - gdcm::SequenceOfItems, 623
 - gdcm::StreamImageReader, 655
 - gdcm::SurfaceReader, 681
 - gdcm::TableReader, 694
 - gdcm::Tag, 700
 - gdcm::UNExplicitDataElement, 787
 - gdcm::UNExplicitImplicitDataElement, 789
 - gdcm::ValueIO, 798
 - gdcm::VL, 801
 - gdcm::VR, 809
 - gdcm::VR16ExplicitDataElement, 811
 - gdcm::VRVLSize< 0 >, 812
 - gdcm::VRVLSize< 1 >, 813
 - itk::GDCMImageIO2, 383
- Read16
 - gdcm::VL, 801
- ReadACRNEMAImage
 - gdcm::ImageReader, 425
 - gdcm::PixmapReader, 543
- ReadCompat
 - gdcm::FileMetaInformation, 361
- ReadCompatInternal
 - gdcm::FileMetaInformation, 361
- ReadComputeLength
 - gdcm::EncodingImplementation< VR::VRASCII >, 337
 - gdcm::EncodingImplementation< VR::VRBINARY >, 338
- ReadDataSet
 - gdcm::Reader, 589
- ReadFiles
 - vtkGDCMThreadedImageReader, 832
- ReadFromCommaSeparatedString
 - gdcm::PrivateTag, 568
 - gdcm::Tag, 700
- ReadFromPipeSeparatedString
 - gdcm::Tag, 700
- ReadImage
 - gdcm::ImageReader, 425

- gdcmm::PixmapReader, 543
- ReadImageInformation
 - gdcmm::StreamImageReader, 655
 - itk::GDCMImageIO2, 383
- ReadImageSubregionJpegLS
 - gdcmm::StreamImageReader, 656
- ReadImageSubregionRAW
 - gdcmm::StreamImageReader, 656
- ReadInto
 - gdcmm::network::PDataTFPDU, 522
 - gdcmm::network::PresentationDataValue, 561
- ReadMetalInformation
 - gdcmm::Reader, 590
- ReadNested
 - gdcmm::DataSet, 283
- ReadNoSwap
 - gdcmm::EncodingImplementation< VR::VRASCII >, 337
 - gdcmm::EncodingImplementation< VR::VRBINARY >, 338
- ReadOrSkip
 - gdcmm::DataElement, 272
- ReadPointMacro
 - gdcmm::SurfaceReader, 681
- ReadPreValue
 - gdcmm::CP246ExplicitDataElement, 245
 - gdcmm::DataElement, 272
 - gdcmm::ExplicitDataElement, 346
 - gdcmm::ExplicitImplicitDataElement, 348
 - gdcmm::ImplicitDataElement, 433
 - gdcmm::UNExplicitDataElement, 787
 - gdcmm::UNExplicitImplicitDataElement, 789
 - gdcmm::VR16ExplicitDataElement, 812
- ReadPreamble
 - gdcmm::Reader, 590
- ReadSegment
 - gdcmm::SegmentReader, 612
- ReadSegments
 - gdcmm::SegmentReader, 612
- ReadSelectedTags
 - gdcmm::DataSet, 283
 - gdcmm::Reader, 590
- ReadSelectedTagsWithLength
 - gdcmm::DataSet, 283
- ReadSurface
 - gdcmm::SurfaceReader, 681
- ReadSurfaces
 - gdcmm::SurfaceReader, 681
- ReadUpToTag
 - gdcmm::DataSet, 283
 - gdcmm::Reader, 590
- ReadUpToTagWithLength
 - gdcmm::DataSet, 283
- ReadVM
 - gdcmm::DictConverter, 300
- ReadVR
 - gdcmm::DictConverter, 300
- ReadValue
 - gdcmm::CP246ExplicitDataElement, 245
 - gdcmm::DataElement, 272
 - gdcmm::ExplicitDataElement, 346
 - gdcmm::ExplicitImplicitDataElement, 348
 - gdcmm::Fragment, 376
 - gdcmm::ImplicitDataElement, 433
 - gdcmm::UNExplicitDataElement, 787
 - gdcmm::UNExplicitImplicitDataElement, 789
 - gdcmm::VR16ExplicitDataElement, 812
- ReadWithLength
 - gdcmm::CP246ExplicitDataElement, 245
 - gdcmm::DataElement, 272
 - gdcmm::DataSet, 283
 - gdcmm::ExplicitDataElement, 346
 - gdcmm::ExplicitImplicitDataElement, 348
 - gdcmm::ImplicitDataElement, 433
 - gdcmm::UNExplicitDataElement, 787
 - gdcmm::VR16ExplicitDataElement, 812
- Reader
 - gdcmm::Reader, 589
- Readuint16
 - gdcmm::DictConverter, 300
- RealWorldValueMappingStorage
 - gdcmm::UIDs, 728
- RecommendedDisplayCIELabToRGB
 - gdcmm::SurfaceHelper, 678
- RecurseDataSet
 - gdcmm::Anonymizer, 150
- red
 - gdcmm::terminal, 130
- reference
 - gdcmm::CodeString, 233
 - gdcmm::LO, 466
 - gdcmm::String, 664
- ReferenceFrameOfReferenceUID
 - vtkRTStructSetProperties, 853
- ReferenceSeriesInstanceUID
 - vtkRTStructSetProperties, 853
- ReferencedColorPrintManagementMetaSOPClassRetired
 - gdcmm::UIDs, 726
- ReferencedGrayscalePrintManagementMetaSOPClass-Retired
 - gdcmm::UIDs, 726
- ReferencedImageBoxSOPClassRetired
 - gdcmm::UIDs, 726
- Register
 - gdcmm::Object, 507
- Remove
 - gdcmm::Anonymizer, 151
 - gdcmm::DataSet, 283

- gdcmm::Preamble, 553
- RemoveAllObservers
 - gdcmm::Subject, 670
- RemoveDictEntry
 - gdcmm::PrivateDict, 566
- RemoveFile
 - gdcmm::System, 690
- RemoveGroupLength
 - gdcmm::Anonymizer, 151
- RemoveObserver
 - gdcmm::Subject, 670
- RemoveOverlay
 - gdcmm::Pixmap, 541
- RemovePrivateTags
 - gdcmm::Anonymizer, 151
- RemoveRetired
 - gdcmm::Anonymizer, 151
- Render
 - vtkImageColorViewer, 839
- RenderWindow
 - vtkImageColorViewer, 842
- Renderer
 - vtkImageColorViewer, 842
- Replace
 - gdcmm::Anonymizer, 151
 - gdcmm::CommandDataSet, 238
 - gdcmm::DataSet, 283
 - gdcmm::FileMetaInformation, 361
- ReplaceEmpty
 - gdcmm::DataSet, 284
- RequestData
 - vtkGDCMPolyDataReader, 825
 - vtkImageMapToColors16, 843
 - vtkImageMapToWindowLevelColors2, 845
 - vtkImagePlanarComponentsToComponents, 846
- RequestData_HemodynamicWaveformStorage
 - vtkGDCMPolyDataReader, 825
- RequestData_RTStructureSetStorage
 - vtkGDCMPolyDataReader, 825
- RequestDataCompat
 - vtkGDCMImageReader, 816
 - vtkGDCMThreadedImageReader, 833
- RequestInformation
 - vtkGDCMPolyDataReader, 826
 - vtkGDCMThreadedImageReader2, 834
 - vtkImageMapToColors16, 843
 - vtkImageMapToWindowLevelColors2, 845
- RequestInformation_HemodynamicWaveformStorage
 - vtkGDCMPolyDataReader, 826
- RequestInformation_RTStructureSetStorage
 - vtkGDCMPolyDataReader, 826
- RequestInformationCompat
 - vtkGDCMImageReader, 816
- RequestPaddedCompositePixelCode
 - gdcmm::ImageCodec, 415
- RequestPlanarConfiguration
 - gdcmm::ImageCodec, 415
- Rescale
 - gdcmm::Rescaler, 593
- RescaleFunctionIntoBestFit
 - gdcmm::Rescaler, 593
- Rescaler
 - gdcmm::Rescaler, 592
- reset
 - gdcmm::terminal, 130
- ResetHandledDataSet
 - gdcmm::network::ULConnectionCallback, 778
- RetrieveSOPInstanceUIDFromIndex
 - gdcmm::DirectoryHelper, 315
- RetrieveSOPInstanceUIDFromZPosition
 - gdcmm::DirectoryHelper, 315
- reverse
 - gdcmm::terminal, 130
- reverse_iterator
 - gdcmm::CodeString, 233
 - gdcmm::LO, 466
 - gdcmm::String, 664
- SAGITTAL
 - gdcmm::Orientation, 509
- SH
 - gdcmm::VR, 808
- SIEMENS
 - gdcmm::Dicts, 306
- SINGLEBIT
 - gdcmm::PixelFormat, 535
- SL
 - gdcmm::VR, 808
- SLICE_ORIENTATION_XY
 - vtkImageColorViewer, 838
- SLICE_ORIENTATION_XZ
 - vtkImageColorViewer, 838
- SLICE_ORIENTATION_YZ
 - vtkImageColorViewer, 838
- SPM2AVG152PDFFrameofReference
 - gdcmm::UIDs, 725
- SPM2AVG152T1FrameofReference
 - gdcmm::UIDs, 725
- SPM2AVG152T2FrameofReference
 - gdcmm::UIDs, 725
- SPM2AVG305T1FrameofReference
 - gdcmm::UIDs, 725
- SPM2BRAINMASKFrameofReference
 - gdcmm::UIDs, 725
- SPM2CSFFFrameofReference
 - gdcmm::UIDs, 725
- SPM2EPIFrameofReference
 - gdcmm::UIDs, 725

- SPM2FILT1FrameofReference
 - gdcm::UIDs, 725
- SPM2GRAYFrameofReference
 - gdcm::UIDs, 725
- SPM2PDFFrameofReference
 - gdcm::UIDs, 725
- SPM2PETFrameofReference
 - gdcm::UIDs, 725
- SPM2SINGLESUBJT1FrameofReference
 - gdcm::UIDs, 725
- SPM2SPECTFrameofReference
 - gdcm::UIDs, 725
- SPM2T1FrameofReference
 - gdcm::UIDs, 725
- SPM2T2FrameofReference
 - gdcm::UIDs, 725
- SPM2TRANSMFrameofReference
 - gdcm::UIDs, 725
- SPM2WHITEFrameofReference
 - gdcm::UIDs, 725
- SQ
 - gdcm::VR, 808
- SS
 - gdcm::VR, 808
- ST
 - gdcm::VR, 808
- STATES_END
 - gdcm::Surface, 673
- SURFACE
 - gdcm::Surface, 674
- SV10
 - gdcm::CSAHeader, 253
- SHA1
 - gdcm::SHA1, 633
- SHComp
 - gdcm, 117
- SOPInstanceUID
 - vtkRTStructSetProperties, 853
- STATES
 - gdcm::Surface, 673
- STComp
 - gdcm, 117
- ScalarType
 - gdcm::PixelFormat, 535
- Scale
 - vtkGDCMImageReader, 819
- Scan
 - gdcm::Scanner, 603
- Scanner
 - gdcm::Scanner, 601
- SecondaryCaptureImageStorage
 - gdcm::MediaStorage, 479
 - gdcm::UIDs, 727
- Segment
 - gdcm::Segment, 606
- SegmentAlgorithmName
 - gdcm::Segment, 607
- SegmentAlgorithmType
 - gdcm::Segment, 607
- SegmentDescription
 - gdcm::Segment, 607
- SegmentLabel
 - gdcm::Segment, 607
- SegmentMap
 - gdcm::SegmentReader, 611
- SegmentNumber
 - gdcm::Segment, 607
- SegmentReader
 - gdcm::SegmentReader, 611
- SegmentVector
 - gdcm::SegmentReader, 611
 - gdcm::SegmentWriter, 614
- SegmentWriter
 - gdcm::SegmentWriter, 614
- Segmentation
 - gdcm::MediaStorage, 481
- SegmentationStorage
 - gdcm::MediaStorage, 481
 - gdcm::UIDs, 728
- SegmentedPaletteColorLookupTable
 - gdcm::SegmentedPaletteColorLookupTable, 609
- Segments
 - gdcm::SegmentReader, 612
 - gdcm::SegmentWriter, 614
- Selection
 - gdcm::Sorter, 646
- SelectionMap
 - gdcm::Sorter, 645
- Self
 - gdcm::AnonymizeEvent, 146
 - gdcm::DataEvent, 277
 - gdcm::DataSetEvent, 286
 - gdcm::MemberCommand, 485
 - gdcm::ProgressEvent, 570
 - gdcm::SimpleMemberCommand, 636
 - itk::GDCMImageIO2, 379
- SendEcho
 - gdcm::network::ULConnectionManager, 782
 - gdcm::ServiceClassUser, 630
- SendFind
 - gdcm::network::ULConnectionManager, 782
 - gdcm::ServiceClassUser, 631
- SendMove
 - gdcm::network::ULConnectionManager, 782
 - gdcm::ServiceClassUser, 631
- SendStore
 - gdcm::network::ULConnectionManager, 782
 - gdcm::ServiceClassUser, 631

- Separator
 - gdcm::ApplicationEntity, 155
 - gdcm::PersonName, 530
- SequenceLengthField
 - gdcm::SequenceOfItems, 624
- SequenceOfFragments
 - gdcm::SequenceOfFragments, 617
- SequenceOfItems
 - gdcm::SequenceOfItems, 622
- SerieHelper
 - gdcm::SerieHelper, 626
- SerieRestrictions
 - gdcm::SerieHelper, 626
- Series
 - gdcm::Series, 628
- SeriesInstanceUID
 - vtkRTStructSetProperties, 853
- ServiceClassUser
 - gdcm::ServiceClassUser, 630
- Set
 - gdcm::Attribute, 167
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcm::Element, 320
 - gdcm::Element< TVR, VM::VM1_n >, 325
- SetAETitle
 - gdcm::ServiceClassUser, 631
- SetAbstractSyntax
 - gdcm::network::PresentationContextRQ, 560
 - gdcm::PresentationContext, 555
- SetAlgorithmFamily
 - gdcm::Surface, 676
- SetAlgorithmName
 - gdcm::Surface, 676
- SetAlgorithmVersion
 - gdcm::Surface, 676
- SetAnatomicRegion
 - gdcm::Segment, 607
- SetArray
 - gdcm::Element< TVR, VM::VM1_n >, 325
- SetAxisOfRotation
 - gdcm::Surface, 676
- SetBitPosition
 - gdcm::Overlay, 514
- SetBitSample
 - gdcm::JPEGCodec, 460
- SetBitsAllocated
 - gdcm::Overlay, 514
 - gdcm::PixelFormat, 537
- SetBitsStored
 - gdcm::PixelFormat, 537
- SetBlob
 - gdcm::ApplicationEntity, 155
 - gdcm::network::PresentationDataValue, 561
- gdcm::PersonName, 530
- SetBlueLUT
 - gdcm::LookupTable, 471
- SetBufferLength
 - gdcm::JPEGLSCodec, 463
 - gdcm::PNMCodec, 551
 - gdcm::RLECodec, 596
- SetByteSwapTag
 - gdcm::ByteSwapFilter, 214
- SetByteValue
 - gdcm::Attribute, 167
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 179
 - gdcm::CSAElement, 250
 - gdcm::DataElement, 272
- SetByteValueNoSwap
 - gdcm::Attribute, 167
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
- SetCallbackFunction
 - gdcm::MemberCommand, 486
 - gdcm::SimpleMemberCommand, 637
- SetCalledAETitle
 - gdcm::network::AAssociateACPDU, 137
 - gdcm::network::AAssociateRQPDU, 142
 - gdcm::ServiceClassUser, 631
- SetCallingAETitle
 - gdcm::network::AAssociateACPDU, 137
 - gdcm::network::AAssociateRQPDU, 142
- SetCenterOfRotation
 - gdcm::Surface, 676
- SetChangePrivateTags
 - gdcm::FileExplicitFilter, 356
- SetCheckFileMetaInformation
 - gdcm::Writer, 858
- SetCipherType
 - gdcm::CryptographicMessageSyntax, 247
- SetColor
 - gdcm::Printer, 565
- SetColorLevel
 - vtkImageColorViewer, 839
- SetColorWindow
 - vtkImageColorViewer, 839
- SetColumns
 - gdcm::Bitmap, 206
 - gdcm::Overlay, 514
- SetCommand
 - gdcm::network::PresentationDataValue, 561
- SetComponents
 - gdcm::PersonName, 530
- SetCompressIconImage
 - gdcm::ImageChangeTransferSyntax, 408

- SetComputeZSpacing
 - gdcm::IPPSorter, 442
- SetCoordinateStartValue
 - gdcm::Curve, 265
- SetCoordinateStepValue
 - gdcm::Curve, 265
- SetCryptographicMessageSyntax
 - gdcm::Anonymizer, 151
- SetCurve
 - gdcm::Curve, 265
 - vtkGDCMImageReader, 816
- SetCurveDataDescriptor
 - gdcm::Curve, 265
- SetCurveDescription
 - gdcm::Curve, 265
- SetData
 - gdcm::DataEvent, 277
- SetDataElement
 - gdcm::Bitmap, 206
- SetDataSet
 - gdcm::File, 352
 - gdcm::network::PresentationDataValue, 561
- SetDataSetTransferSyntax
 - gdcm::FileMetaInformation, 361
- SetDataValueRepresentation
 - gdcm::Curve, 265
- SetDebug
 - gdcm::Trace, 708
- SetDefaultTransferSyntax
 - gdcm::PresentationContextGenerator, 558
- SetDerivationCodeSequenceCodeValue
 - gdcm::FileDerivation, 354
- SetDerivationDescription
 - gdcm::FileDerivation, 354
- SetDescription
 - gdcm::CSAHeaderDictEntry, 259
 - gdcm::ModuleEntry, 495
 - gdcm::Overlay, 514
- SetDescriptor
 - gdcm::DICOMDIRGenerator, 296
- SetDictName
 - gdcm::DictConverter, 300
- SetDicts
 - gdcm::PythonFilter, 574
 - gdcm::StringFilter, 667
- SetDimension
 - gdcm::Bitmap, 206
- SetDimensions
 - gdcm::Bitmap, 207
 - gdcm::Curve, 265
 - gdcm::ImageCodec, 414
- SetDimensionsValue
 - gdcm::ImageHelper, 421
- SetDirectionCosines
 - gdcm::Image, 396
 - vtkGDCMImageWriter, 821
- SetDirectionCosinesFromImageOrientationPatient
 - vtkGDCMImageWriter, 821
- SetDirectionCosinesTolerance
 - gdcm::IPPSorter, 442
- SetDirectionCosinesValue
 - gdcm::ImageHelper, 421
- SetDirectory
 - gdcm::network::ULWritingCallback, 785
 - gdcm::SerieHelper, 627
- SetDisplayId
 - vtkImageColorViewer, 839
- SetElement
 - gdcm::Tag, 700
- SetElementHandler
 - gdcm::Parser, 519
- SetElementTag
 - gdcm::Tag, 700
- SetElementXX
 - gdcm::DictEntry, 303
- SetError
 - gdcm::Trace, 708
- SetEvent
 - gdcm::network::ULEvent, 783
- SetFile
 - gdcm::Anonymizer, 151
 - gdcm::DICOMDIRGenerator, 296
 - gdcm::FileDerivation, 354
 - gdcm::FileExplicitFilter, 356
 - gdcm::IconImageFilter, 390
 - gdcm::Printer, 565
 - gdcm::PythonFilter, 574
 - gdcm::Reader, 590
 - gdcm::SplitMosaicFilter, 650
 - gdcm::StreamImageWriter, 660
 - gdcm::StringFilter, 667
 - gdcm::Validate, 795
 - gdcm::Writer, 858
- SetFileName
 - gdcm::Reader, 590
 - gdcm::StreamImageReader, 656
 - gdcm::StreamImageWriter, 660
 - gdcm::Writer, 859
 - vtkGDCMThreadedImageReader2, 834
- SetFileNames
 - vtkGDCMImageReader, 816
 - vtkGDCMImageWriter, 821
 - vtkGDCMThreadedImageReader2, 834
- SetFilePattern
 - vtkGDCMImageReader, 816
- SetFilePrefix
 - vtkGDCMImageReader, 816
- SetFilename

- gdcmm::TableReader, 694
- SetFilenames
 - gdcmm::DICOMDIRGenerator, 296
- SetFiles
 - gdcmm::FileSet, 367
- SetFiniteVolume
 - gdcmm::Surface, 676
- SetForce
 - gdcmm::ImageChangeTransferSyntax, 409
 - gdcmm::ImageFragmentSplitter, 418
- SetForcePixelSpacing
 - gdcmm::ImageHelper, 421
- SetForceRescaleInterceptSlope
 - gdcmm::ImageHelper, 421
- SetFragmentSizeMax
 - gdcmm::ImageFragmentSplitter, 418
- SetFrameOrigin
 - gdcmm::Overlay, 514
- SetFromDataElement
 - gdcmm::Attribute, 167
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 179
 - gdcmm::Element, 320
 - gdcmm::Element< TVR, VM::VM1_n >, 325
- SetFromDataSet
 - gdcmm::Attribute, 168
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 173
 - gdcmm::MediaStorage, 482
- SetFromFile
 - gdcmm::MediaStorage, 482
- SetFromHeader
 - gdcmm::MediaStorage, 483
- SetFromModality
 - gdcmm::MediaStorage, 483
- SetFromSourceImageSequence
 - gdcmm::MediaStorage, 483
- SetFromString
 - gdcmm::DirectionCosines, 311
- SetFromUID
 - gdcmm::UIDs, 738
- SetGreenLUT
 - gdcmm::LookupTable, 471
- SetGroup
 - gdcmm::Curve, 265
 - gdcmm::Overlay, 514
 - gdcmm::Tag, 700
- SetGroupXX
 - gdcmm::DictEntry, 303
- SetHeader
 - gdcmm::File, 352
- SetHighBit
 - gdcmm::PixelFormat, 537
- SetHostname
 - gdcmm::ServiceClassUser, 631
- SetIE
 - gdcmm::IODEntry, 438
- SetIconImage
 - gdcmm::Pixmap, 541
- SetImage
 - gdcmm::PixmapWriter, 549
 - gdcmm::SplitMosaicFilter, 650
- SetImplementationClassUID
 - gdcmm::FileMetaInformation, 361
- SetImplementationVersionName
 - gdcmm::FileMetaInformation, 361
- SetInput
 - gdcmm::BitmapToBitmapFilter, 211
 - gdcmm::ImageConverter, 416
 - vtkImageColorViewer, 839
- SetInputConnection
 - vtkImageColorViewer, 840
- SetInputFileName
 - gdcmm::DictConverter, 300
- SetIntercept
 - gdcmm::Image, 396
 - gdcmm::Rescaler, 593
- SetKey
 - gdcmm::CSAElement, 250
- SetKeyword
 - gdcmm::DictEntry, 303
- SetLUT
 - gdcmm::Bitmap, 207
 - gdcmm::ImageCodec, 414
 - gdcmm::LookupTable, 471
 - gdcmm::SegmentedPaletteColorLookupTable, 609
- SetLastElement
 - gdcmm::ParseException, 517
- SetLastFragment
 - gdcmm::network::PresentationDataValue, 561
- SetLength
 - gdcmm::ByteValue, 218
 - gdcmm::Element< TVR, VM::VM1_2 >, 322
 - gdcmm::Element< TVR, VM::VM1_n >, 325
 - gdcmm::Element< TVR, VM::VM2_2n >, 327
 - gdcmm::Element< TVR, VM::VM2_n >, 329
 - gdcmm::Element< TVR, VM::VM3_3n >, 330
 - gdcmm::Element< TVR, VM::VM3_n >, 332
 - gdcmm::RLECodec, 596
 - gdcmm::SequenceOfFragments, 618
 - gdcmm::SequenceOfItems, 623
 - gdcmm::Value, 797
- SetLengthToUndefined
 - gdcmm::SequenceOfItems, 624
- SetLoadMode
 - gdcmm::SerieHelper, 627

- SetLoadPrivateTagsDefault
 - itk::GDCMImageIO2, 383
- SetLoadSequencesDefault
 - itk::GDCMImageIO2, 383
- SetLookupTable
 - vtkImageMapToColors16, 843
- SetLossless
 - gdcm::JPEGCodec, 460
 - gdcm::JPEGLSCodec, 463
- SetLossyError
 - gdcm::JPEGLSCodec, 463
- SetLossyFlag
 - gdcm::Bitmap, 207
 - gdcm::ImageCodec, 414
- SetManifold
 - gdcm::Surface, 676
- SetMaxPDULength
 - gdcm::network::ULConnectionInfo, 779
- SetMaxPDUSize
 - gdcm::network::ULConnection, 777
- SetMaximumLength
 - gdcm::network::MaximumLengthSub, 475
- SetMaximumPointDistance
 - gdcm::Surface, 676
- SetMeanPointDistance
 - gdcm::Surface, 676
- SetMedicalImageProperties
 - vtkGDCMImageReader, 816
 - vtkGDCMImageWriter, 821
 - vtkGDCMPolyDataWriter, 828
- SetMergeModeToAbstractSyntax
 - gdcm::PresentationContextGenerator, 558
- SetMergeModeToTransferSyntax
 - gdcm::PresentationContextGenerator, 558
- SetMeshPrimitive
 - gdcm::Surface, 676
- SetMessageHeader
 - gdcm::network::PresentationDataValue, 561
- SetMinMaxForPixelType
 - gdcm::Rescaler, 593
- SetName
 - gdcm::CSAElement, 250
 - gdcm::CSAHeaderDictEntry, 259
 - gdcm::DictEntry, 303
 - gdcm::IODEntry, 438
 - gdcm::Macro, 473
 - gdcm::Module, 493
 - gdcm::ModuleEntry, 495
 - gdcm::network::AbstractSyntax, 144
 - gdcm::network::ApplicationContext, 154
 - gdcm::network::TransferSyntaxSub, 713
 - gdcm::PDBElement, 523
- SetNameFromUID
 - gdcm::network::AbstractSyntax, 144
- gdcm::network::TransferSyntaxSub, 713
- SetNeedByteSwap
 - gdcm::Bitmap, 207
 - gdcm::ImageCodec, 414
- SetNeedOverlayCleanup
 - gdcm::ImageCodec, 414
- SetNestedDataSet
 - gdcm::Item, 446
- SetNoOfItems
 - gdcm::CSAElement, 250
- SetNoSwap
 - gdcm::Element, 320
 - gdcm::Element< TVR, VM::VM1_n >, 325
- SetNumberOfCurves
 - gdcm::Pixmap, 541
- SetNumberOfDimensions
 - gdcm::Bitmap, 207
 - gdcm::ImageCodec, 414
- SetNumberOfFilenames
 - gdcm::FilenameGenerator, 366
- SetNumberOfFrames
 - gdcm::Overlay, 514
- SetNumberOfInputPorts
 - vtkGDCMPolyDataWriter, 828
- SetNumberOfItems
 - gdcm::SequenceOfItems, 624
- SetNumberOfOverlays
 - gdcm::Pixmap, 541
- SetNumberOfPoints
 - gdcm::Curve, 265
- SetNumberOfResolutions
 - gdcm::JPEG2000Codec, 454
- SetNumberOfSegments
 - gdcm::SegmentWriter, 614
- SetNumberOfSurfacePoints
 - gdcm::Surface, 676
- SetNumberOfSurfaces
 - gdcm::SurfaceWriter, 683
- SetNumberOfTableValues
 - vtkLookupTable16, 849
- SetNumberOfValues
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 179
- SetNumberOfVectors
 - gdcm::Surface, 676
- SetObliquityThresholdCosineValue
 - gdcm::Orientation, 510
- SetOffScreenRendering
 - vtkImageColorViewer, 840
- SetOrigin
 - gdcm::Image, 396
 - gdcm::Overlay, 515
- SetOriginValue
 - gdcm::ImageHelper, 421

- SetOutputDimensions
 - gdcm::IconImageGenerator, 392
- SetOutputFileName
 - gdcm::DictConverter, 300
- SetOutputFormatToLuminance
 - vtkImageMapToColors16, 843
- SetOutputFormatToLuminanceAlpha
 - vtkImageMapToColors16, 843
- SetOutputFormatToRGB
 - vtkImageMapToColors16, 843
- SetOutputFormatToRGBA
 - vtkImageMapToColors16, 843
- SetOutputType
 - gdcm::DictConverter, 300
- SetOutsideValuePixel
 - gdcm::IconImageGenerator, 392
- SetOverlay
 - gdcm::Overlay, 515
- SetOverlayVisibility
 - vtkImageColorViewer, 840
- SetOwner
 - gdcm::PrivateTag, 568
- SetPDU
 - gdcm::network::ULEvent, 783
- SetParentId
 - vtkImageColorViewer, 840
- SetPattern
 - gdcm::FilenameGenerator, 366
- SetPermissions
 - gdcm::System, 690
- SetPhotometricInterpretation
 - gdcm::Bitmap, 207
 - gdcm::ImageChangePhotometricInterpretation, 402
 - gdcm::ImageCodec, 414
- SetPixelFormat
 - gdcm::Bitmap, 207
 - gdcm::ImageCodec, 414
 - gdcm::JPEGCodec, 460
 - gdcm::Rescaler, 593
- SetPixelMinMax
 - gdcm::IconImageGenerator, 392
- SetPixelRepresentation
 - gdcm::PixelFormat, 537
- SetPixmap
 - gdcm::IconImageGenerator, 392
 - gdcm::PixmapWriter, 549
- SetPlanarConfiguration
 - gdcm::Bitmap, 207
 - gdcm::ImageChangePlanarConfiguration, 405
 - gdcm::ImageCodec, 414
- SetPointCoordinatesData
 - gdcm::Surface, 676
- SetPointPositionAccuracy
 - gdcm::Surface, 676
- SetPointsBoundingBoxCoordinates
 - gdcm::Surface, 676
- SetPort
 - gdcm::ServiceClassUser, 632
- SetPortSCP
 - gdcm::ServiceClassUser, 632
- SetPosition
 - vtkImageColorViewer, 840
- SetPreamble
 - gdcm::FileMetaInformation, 361
- SetPrefix
 - gdcm::FilenameGenerator, 366
- SetPresentationContextId
 - gdcm::network::PresentationContextAC, 555
 - gdcm::network::PresentationContextRQ, 560
 - gdcm::network::PresentationDataValue, 561
 - gdcm::PresentationContext, 555
- SetPresentationContexts
 - gdcm::network::ULConnection, 777
 - gdcm::ServiceClassUser, 632
- SetPrimitiveData
 - gdcm::MeshPrimitive, 490
- SetPrimitiveType
 - gdcm::MeshPrimitive, 490
- SetPrimitivesData
 - gdcm::MeshPrimitive, 490
- SetPrivateCreator
 - gdcm::Tag, 700
- SetProcessingAlgorithm
 - gdcm::Surface, 676
- SetProgress
 - gdcm::ProgressEvent, 570
- SetPropertyCategory
 - gdcm::Segment, 607
- SetPropertyType
 - gdcm::Segment, 607
- SetPurposeOfReferenceCodeSequenceCodeValue
 - gdcm::FileDerivation, 354
- SetQuality
 - gdcm::JPEG2000Codec, 454
 - gdcm::JPEGCodec, 460
- SetRTStructSetProperties
 - vtkGDCMPolyDataWriter, 828
- SetRate
 - gdcm::JPEG2000Codec, 454
- SetRecommendedDisplayCIELabValue
 - gdcm::Surface, 676
- SetRecommendedDisplayGrayscaleValue
 - gdcm::Surface, 676
- SetRecommendedPresentationOpacity
 - gdcm::Surface, 676
- SetRecommendedPresentationType
 - gdcm::Surface, 677
- SetRecomputeItemLength

- gdcm::FileExplicitFilter, 356
- SetRecomputeSequenceLength
 - gdcm::FileExplicitFilter, 356
- SetRedLUT
 - gdcm::LookupTable, 471
- SetRef
 - gdcm::IODEntry, 438
- SetRenderWindow
 - vtkImageColorViewer, 840
- SetRenderer
 - vtkImageColorViewer, 840
- SetRescaleInterceptSlopeValue
 - gdcm::ImageHelper, 421
- SetRetired
 - gdcm::DictEntry, 303
- SetReversible
 - gdcm::JPEG2000Codec, 454
- SetRoot
 - gdcm::UIDGenerator, 719
- SetRootDirectory
 - gdcm::DICOMDIRGenerator, 296
- SetRows
 - gdcm::Bitmap, 207
 - gdcm::Overlay, 515
- SetSamplesPerPixel
 - gdcm::PixelFormat, 537
- SetScalarType
 - gdcm::PixelFormat, 537
- SetSearchParameter
 - gdcm::BaseRootQuery, 195
- SetSegmentAlgorithmName
 - gdcm::Segment, 607
- SetSegmentAlgorithmType
 - gdcm::Segment, 607
- SetSegmentDescription
 - gdcm::Segment, 607
- SetSegmentLabel
 - gdcm::Segment, 607
- SetSegmentNumber
 - gdcm::Segment, 607
- SetSegments
 - gdcm::SegmentWriter, 614
- SetSize
 - vtkImageColorViewer, 840
- SetSlice
 - vtkImageColorViewer, 840
- SetSliceOrientation
 - vtkImageColorViewer, 840
- SetSliceOrientationToXY
 - vtkImageColorViewer, 840
- SetSliceOrientationToXZ
 - vtkImageColorViewer, 840
- SetSliceOrientationToYZ
 - vtkImageColorViewer, 840
- SetSlope
 - gdcm::Image, 396
 - gdcm::Rescaler, 593
- SetSortFunction
 - gdcm::Sorter, 646
- SetSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, 361
- SetSpacing
 - gdcm::Image, 396
- SetSpacingValue
 - gdcm::ImageHelper, 421
- SetState
 - gdcm::network::ULConnection, 777
- SetStream
 - gdcm::Reader, 590
 - gdcm::StreamImageReader, 656
 - gdcm::StreamImageWriter, 660
 - gdcm::Trace, 708
 - gdcm::Writer, 859
- SetStyle
 - gdcm::Printer, 565
- SetSurfaceComments
 - gdcm::Surface, 677
- SetSurfaceCount
 - gdcm::Segment, 607
- SetSurfaceNumber
 - gdcm::Surface, 677
- SetSurfaceProcessing
 - gdcm::Surface, 677
- SetSurfaceProcessingDescription
 - gdcm::Surface, 677
- SetSurfaceProcessingRatio
 - gdcm::Surface, 677
- SetSyngoDT
 - gdcm::CSAElement, 250
- SetTag
 - gdcm::AnonymizeEvent, 146
 - gdcm::DataElement, 273
- SetTargetPixelType
 - gdcm::Rescaler, 593
- SetTileSize
 - gdcm::JPEG2000Codec, 454
- SetTimeout
 - gdcm::network::ARTIMTimer, 160
 - gdcm::ServiceClassUser, 632
- SetToUndefined
 - gdcm::VL, 801
- SetTransferSyntax
 - gdcm::Bitmap, 208
 - gdcm::ImageChangeTransferSyntax, 409
 - gdcm::network::PresentationContextAC, 556
- SetType
 - gdcm::ModuleEntry, 495
 - gdcm::Overlay, 515

- SetTypeOfData
 - gdcm::Curve, 265
- SetUsage
 - gdcm::IODEntry, 438
- SetUseSeriesDetails
 - gdcm::SerieHelper, 627
- SetUseTargetPixelType
 - gdcm::Rescaler, 593
- SetUseVRUN
 - gdcm::FileExplicitFilter, 356
- SetUserCodec
 - gdcm::ImageChangeTransferSyntax, 409
- SetUserData
 - gdcm::Parser, 519
- SetVL
 - gdcm::DataElement, 273
- SetVLToUndefined
 - gdcm::DataElement, 273
- SetVM
 - gdcm::CSAElement, 250
 - gdcm::CSAHeaderDictEntry, 259
 - gdcm::DictEntry, 303
- SetVR
 - gdcm::CSAElement, 250
 - gdcm::CSAHeaderDictEntry, 259
 - gdcm::DataElement, 273
 - gdcm::DictEntry, 303
- SetValue
 - gdcm::Attribute, 168
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 179
 - gdcm::CSAElement, 250
 - gdcm::DataElement, 273
 - gdcm::Element, 320
 - gdcm::Element< TVR, VM::VM1_n >, 325
 - gdcm::PDBelement, 524
- SetValues
 - gdcm::Attribute, 168
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 180
- SetVectorAccuracy
 - gdcm::Surface, 677
- SetVectorCoordinateData
 - gdcm::Surface, 677
- SetVectorDimensionality
 - gdcm::Surface, 677
- SetWarning
 - gdcm::Trace, 708
- SetWindowId
 - vtkImageColorViewer, 841
- SetWriteDataSetOnly
 - gdcm::Writer, 859
- SetZSpacingTolerance
 - gdcm::IPPSorter, 442
- setAttribute
 - gdcm::terminal, 131
- setbgcolor
 - gdcm::terminal, 131
- setfgcolor
 - gdcm::terminal, 131
- setmode
 - gdcm::terminal, 131
- SetupInteractor
 - vtkImageColorViewer, 841
- Shift
 - vtkGDCMImageReader, 819
- ShiftEnd
 - gdcm::ByteBuffer, 212
- ShowAbort
 - gdcm::SimpleSubjectWatcher, 638
- ShowAnonymization
 - gdcm::SimpleSubjectWatcher, 638
- ShowData
 - gdcm::SimpleSubjectWatcher, 638
- ShowDataSet
 - gdcm::SimpleSubjectWatcher, 638
- ShowIteration
 - gdcm::SimpleSubjectWatcher, 638
- ShowProgress
 - gdcm::SimpleSubjectWatcher, 638
- SimpleMemberCommand
 - gdcm::SimpleMemberCommand, 636
- SimpleSubjectWatcher
 - gdcm::SimpleSubjectWatcher, 638
- SingleSerieUIDFileSetHT
 - gdcm::SerieHelper, 627
- SingleSerieUIDFileSetmap
 - gdcm::SerieHelper, 626
- Size
 - gdcm::CodeString, 234
 - gdcm::DataSet, 284
 - gdcm::GroupDict, 387
 - gdcm::network::AAAbortPDU, 134
 - gdcm::network::AAAssociateACPDU, 137
 - gdcm::network::AAAssociateRJPDU, 139
 - gdcm::network::AAAssociateRQPDU, 142
 - gdcm::network::AbstractSyntax, 144
 - gdcm::network::ApplicationContext, 154
 - gdcm::network::AReleaseRPPDU, 157
 - gdcm::network::AReleaseRQPDU, 159
 - gdcm::network::AsynchronousOperationsWindow-Sub, 162
 - gdcm::network::BasePDU, 192
 - gdcm::network::ImplementationClassUIDSub, 430
 - gdcm::network::ImplementationVersionNameSub, 431

- gdcmm::network::MaximumLengthSub, 475
- gdcmm::network::PDataTFPDU, 522
- gdcmm::network::PresentationContextAC, 556
- gdcmm::network::PresentationContextRQ, 560
- gdcmm::network::PresentationDataValue, 562
- gdcmm::network::TransferSyntaxSub, 713
- gdcmm::network::UserInformation, 794
- size_type
 - gdcmm::CodeString, 233
 - gdcmm::LO, 466
 - gdcmm::String, 664
- SizeType
 - gdcmm::DataSet, 280
 - gdcmm::FilenameGenerator, 365
 - gdcmm::IOD, 435
 - gdcmm::NestedModuleEntries, 504
 - gdcmm::network::AAAssociateACPDU, 137
 - gdcmm::network::AAAssociateRQPDU, 141
 - gdcmm::network::PDataTFPDU, 521
 - gdcmm::network::PresentationContextRQ, 559
 - gdcmm::PresentationContext, 554
 - gdcmm::PresentationContextGenerator, 557
 - gdcmm::SequenceOfFragments, 617
 - gdcmm::SequenceOfItems, 622
- Slice
 - vtkImageColorViewer, 842
- SliceOrientation
 - vtkImageColorViewer, 842
- SmartPointer
 - gdcmm::Object, 507
 - gdcmm::SmartPointer, 641
- Sort
 - gdcmm::IPPSorter, 442
 - gdcmm::Sorter, 646
- SortFunc
 - gdcmm::Sorter, 646
- SortFunction
 - gdcmm::Sorter, 645
- Sorter
 - gdcmm::Sorter, 645
- SpacialFiducialsStorage
 - gdcmm::MediaStorage, 480
- SpacialRegistrationStorage
 - gdcmm::MediaStorage, 480
- Spacing
 - gdcmm::Spacing, 648
- SpacingType
 - gdcmm::Spacing, 648
- SpatialFiducialsStorage
 - gdcmm::UIDs, 728
- SpatialRegistrationStorage
 - gdcmm::UIDs, 728
- Spectroscopy
 - gdcmm::Spectroscopy, 649
- Split
 - gdcmm::ImageFragmentSplitter, 418
 - gdcmm::SplitMosaicFilter, 650
- SplitExtent
 - vtkGDCMThreadedImageReader2, 834
- SplitMosaicFilter
 - gdcmm::SplitMosaicFilter, 650
- Squeeze
 - gdcmm::ApplicationEntity, 155
- StableSort
 - gdcmm::Sorter, 646
- StandaloneCurveStorage
 - gdcmm::MediaStorage, 480
- StandaloneCurveStorageRetired
 - gdcmm::UIDs, 727
- StandaloneModalityLUTStorage
 - gdcmm::MediaStorage, 480
- StandaloneModalityLUTStorageRetired
 - gdcmm::UIDs, 728
- StandaloneOverlayStorage
 - gdcmm::MediaStorage, 480
- StandaloneOverlayStorageRetired
 - gdcmm::UIDs, 727
- StandalonePETCurveStorageRetired
 - gdcmm::UIDs, 729
- StandaloneVOILUTStorage
 - gdcmm::MediaStorage, 480
- StandaloneVOILUTStorageRetired
 - gdcmm::UIDs, 728
- Start
 - gdcmm::network::ARTIMTimer, 160
- StartAssociation
 - gdcmm::ServiceClassUser, 632
- StartElement
 - gdcmm::TableReader, 694
 - gdcmm::XMLDictReader, 861
 - gdcmm::XMLPrivateDictReader, 863
- StartElementHandler
 - gdcmm::Parser, 518
- StartFilter
 - gdcmm::SimpleSubjectWatcher, 638
- StereometricRelationshipStorage
 - gdcmm::UIDs, 728
- Stop
 - gdcmm::network::ARTIMTimer, 160
- StopAssociation
 - gdcmm::ServiceClassUser, 632
- StopProtocol
 - gdcmm::network::ULConnection, 777
- StorageCommitmentPullModelSOPClassRetired
 - gdcmm::UIDs, 726
- StorageCommitmentPullModelSOPInstanceRetired
 - gdcmm::UIDs, 726
- StorageCommitmentPushModelSOPClass

- gdcmm::UIDs, 726
- StorageCommitmentPushModelSOPInstance
 - gdcmm::UIDs, 726
- StorageServiceClass
 - gdcmm::UIDs, 726
- StoredPrintStorageSOPClassRetired
 - gdcmm::UIDs, 727
- StrCaseCmp
 - gdcmm::System, 690
- StrNCaseCmp
 - gdcmm::System, 690
- StrTokR
 - gdcmm::System, 690
- Stream
 - gdcmm::Writer, 859
- StreamImageReader
 - gdcmm::Reader, 590
 - gdcmm::StreamImageReader, 654
- StreamImageWriter
 - gdcmm::StreamImageWriter, 659
 - gdcmm::Writer, 859
- String
 - gdcmm::String, 664, 665
- StringFilter
 - gdcmm::StringFilter, 666
- StructureSetDate
 - vtkRTStructSetProperties, 853
- StructureSetLabel
 - vtkRTStructSetProperties, 853
- StructureSetName
 - vtkRTStructSetProperties, 853
- StructureSetTime
 - vtkRTStructSetProperties, 853
- Study
 - gdcmm::Study, 668
- StudyComponentManagementSOPClass
 - gdcmm::MediaStorage, 480
- StudyComponentManagementSOPClassRetired
 - gdcmm::UIDs, 726
- StudyRootQueryRetrieveInformationModelFIND
 - gdcmm::UIDs, 729
- StudyRootQueryRetrieveInformationModelGET
 - gdcmm::UIDs, 729
- StudyRootQueryRetrieveInformationModelMOVE
 - gdcmm::UIDs, 729
- StudyInstanceUID
 - vtkRTStructSetProperties, 854
- Subject
 - gdcmm::Subject, 669
- SubstanceAdministrationLoggingSOPClass
 - gdcmm::UIDs, 726
- SubstanceAdministrationLoggingSOPInstance
 - gdcmm::UIDs, 726
- SubstanceApprovalQuerySOPClass
 - gdcmm::UIDs, 730
- Superclass
 - gdcmm::AnonymizeEvent, 146
 - gdcmm::DataEvent, 277
 - gdcmm::DataSetEvent, 286
 - gdcmm::LO, 466
 - gdcmm::ProgressEvent, 570
 - itk::GDCMImageIO2, 379
- Surface
 - gdcmm::Surface, 674
- SurfaceSegmentationStorage
 - gdcmm::MediaStorage, 481
- SurfaceCount
 - gdcmm::Segment, 607
- SurfaceReader
 - gdcmm::SurfaceReader, 681
- SurfaceVector
 - gdcmm::Segment, 606
- SurfaceWriter
 - gdcmm::SurfaceWriter, 683
- Surfaces
 - gdcmm::Segment, 607
- Swap
 - gdcmm::ByteSwap, 212
 - gdcmm::SwapperDoOp, 685
 - gdcmm::SwapperNoOp, 686
- SwapArray
 - gdcmm::SwapperDoOp, 685
 - gdcmm::SwapperNoOp, 686
- SwapCode
 - gdcmm::SwapCode, 685
- SwapCodeType
 - gdcmm::SwapCode, 684
- SwapFromSwapCodeIntoSystem
 - gdcmm::ByteSwap, 212
- SwapRange
 - gdcmm::ByteSwap, 213
- SwapRangeFromSwapCodeIntoSystem
 - gdcmm::ByteSwap, 213
- SyngoDTField
 - gdcmm::CSAElement, 251
- SyntaxError
 - gdcmm::Parser, 518
- SystemIsBigEndian
 - gdcmm::ByteSwap, 213
- SystemIsLittleEndian
 - gdcmm::ByteSwap, 213
- T1
 - gdcmm::Type, 716
- T1C
 - gdcmm::Type, 716
- T2
 - gdcmm::Type, 716

- T2C
 - gdcm::Type, 716
- T3
 - gdcm::Type, 716
- TM
 - gdcm::VR, 808
- TRIANGLE
 - gdcm::MeshPrimitive, 489
- TRIANGLE_FAN
 - gdcm::MeshPrimitive, 489
- TRIANGLE_STRIP
 - gdcm::MeshPrimitive, 489
- TS_END
 - gdcm::TransferSyntax, 711
- TCompressionType
 - itk::GDCMImageIO2, 379
- TConstMemberFunctionPointer
 - gdcm::MemberCommand, 485
- TMComp
 - gdcm, 117
- TMemberFunctionPointer
 - gdcm::MemberCommand, 485
 - gdcm::SimpleMemberCommand, 636
- TS
 - gdcm::Bitmap, 209
- TSName
 - gdcm::UIDs, 724
- TSType
 - gdcm::TransferSyntax, 710
 - gdcm::UIDs, 731
- TYPETOENCODING
 - gdcm, 123
 - gdcmVR.h, 945
- TYPETOLENGTH
 - gdcmVM.h, 943
- Table
 - gdcm::Table, 691
- Table16
 - vtkLookupTable16, 850
- TableEntry
 - gdcm::TableEntry, 692
- TableReader
 - gdcm::TableReader, 693
- Tag
 - gdcm::Tag, 697
- tag
 - gdcm::Tag, 701
- TagMismatchError
 - gdcm::Parser, 518
- TagField
 - gdcm::DataElement, 274
- TagPath
 - gdcm::TagPath, 702
- TagToValue
 - gdcm::Scanner, 600
- TagToValueValueType
 - gdcm::Scanner, 600
- tags
 - gdcm::Tag, 701
- TalairachBrainAtlasFrameofReference
 - gdcm::UIDs, 725
- TestAbortOff
 - gdcm::SimpleSubjectWatcher, 638
- TestAbortOn
 - gdcm::SimpleSubjectWatcher, 638
- TestPBKDF2
 - gdcm::ASN1, 161
- Testing
 - gdcm::Testing, 704
- TestsList.txt, 947
- TextSRStorageTrialRetired
 - gdcm::UIDs, 728
- ThreadedExecute
 - vtkImageRGBToYBR, 847
 - vtkImageYBRToRGB, 848
- ThreadedRequestData
 - vtkGDCMThreadedImageReader2, 834
 - vtkImageMapToColors16, 843
 - vtkImageMapToWindowLevelColors2, 845
- to_string
 - gdcm, 123
- ToPyObject
 - gdcm::PythonFilter, 574
- ToString
 - gdcm::StringFilter, 667
- ToStringPair
 - gdcm::StringFilter, 667
- ToUnixSlashes
 - gdcm::Filename, 363
- ToWindowsSlashes
 - gdcm::Filename, 364
- ToshibaPrivateDataStorage
 - gdcm::MediaStorage, 480
- Trace
 - gdcm::Trace, 707
- TransferSyntax
 - gdcm::TransferSyntax, 711
- TransferSyntaxArrayType
 - gdcm::PresentationContext, 554
- TransferSyntaxStringsType
 - gdcm::UIDs, 724
- TransferSyntaxSub
 - gdcm::network::TransferSyntaxSub, 713
- Transition
 - gdcm::network::Transition, 714
- transitions
 - gdcm::network::TableRow, 694
- Trim

- gdcM::String, 665
- TrimInternal
 - gdcM::CodeString, 234
- Truncate
 - gdcM::String, 665
- TryJPEG2000Codec
 - gdcM::Bitmap, 208
 - gdcM::ImageChangeTransferSyntax, 409
- TryJPEG2000Codec2
 - gdcM::Bitmap, 208
- TryJPEGCodec
 - gdcM::Bitmap, 208
 - gdcM::ImageChangeTransferSyntax, 409
- TryJPEGCodec2
 - gdcM::Bitmap, 208
- TryJPEGLSCodec
 - gdcM::Bitmap, 208
 - gdcM::ImageChangeTransferSyntax, 409
- TryKAKADUCodec
 - gdcM::Bitmap, 208
- TryPVRGCodec
 - gdcM::Bitmap, 208
- TryRAWCodec
 - gdcM::Bitmap, 208
 - gdcM::ImageChangeTransferSyntax, 409
- TryRLECodec
 - gdcM::Bitmap, 208
 - gdcM::ImageChangeTransferSyntax, 409
- Type
 - gdcM::Element, 319
 - gdcM::Element< TVR, VM::VM1_n >, 324
 - gdcM::Type, 716
 - gdcM::VL, 800
- TypeType
 - gdcM::Type, 716
- UI
 - gdcM::VR, 808
- UINT12
 - gdcM::PixelFormat, 535
- UINT16
 - gdcM::PixelFormat, 535
- UINT32
 - gdcM::PixelFormat, 535
- UINT8
 - gdcM::PixelFormat, 535
- UL
 - gdcM::VR, 808
- UN
 - gdcM::VR, 808
- UNKNOWN
 - gdcM::PhotometricInterpretation, 532
- UNKNOWN
 - gdcM::CSAHeader, 253
- gdcM::LookupTable, 469
- gdcM::Orientation, 509
- gdcM::PixelFormat, 535
- gdcM::Spacing, 648
- gdcM::Surface, 673
- gdcM::Type, 716
- URI
 - gdcM::MediaStorage, 481
- US
 - gdcM::VR, 808
- US_SS
 - gdcM::VR, 808
- US_SS_OW
 - gdcM::VR, 808
- UT
 - gdcM::VR, 808
- UIComp
 - gdcM, 117
- UIDGenerator
 - gdcM::UIDGenerator, 718
- UINT32_MAX
 - gdcMTypes.h, 935
- ULAction
 - gdcM::network::ULAction, 740
- ULBasicCallback
 - gdcM::network::ULBasicCallback, 774
- ULConnection
 - gdcM::network::ULConnection, 776
- ULConnectionCallback
 - gdcM::network::ULConnectionCallback, 778
- ULConnectionInfo
 - gdcM::network::ULConnectionInfo, 779
- ULConnectionManager
 - gdcM::network::ULConnectionManager, 781
- ULEvent
 - gdcM::network::ULEvent, 783
- ULTransitionTable
 - gdcM::network::ULTransitionTable, 783
- ULWritingCallback
 - gdcM::network::ULWritingCallback, 785
- UTComp
 - gdcM, 117
- uid_1_2_840_10008_15_0_3_1
 - gdcM::UIDs, 736
- uid_1_2_840_10008_15_0_3_10
 - gdcM::UIDs, 736
- uid_1_2_840_10008_15_0_3_11
 - gdcM::UIDs, 736
- uid_1_2_840_10008_15_0_3_12
 - gdcM::UIDs, 736
- uid_1_2_840_10008_15_0_3_13
 - gdcM::UIDs, 736
- uid_1_2_840_10008_15_0_3_14
 - gdcM::UIDs, 736

uid_1_2_840_10008_15_0_3_15
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_16
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_17
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_18
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_19
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_2
gdcm::UIDs, 736

uid_1_2_840_10008_15_0_3_20
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_21
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_22
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_23
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_24
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_25
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_26
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_27
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_28
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_29
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_3
gdcm::UIDs, 736

uid_1_2_840_10008_15_0_3_30
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_31
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_3_4
gdcm::UIDs, 736

uid_1_2_840_10008_15_0_3_5
gdcm::UIDs, 736

uid_1_2_840_10008_15_0_3_6
gdcm::UIDs, 736

uid_1_2_840_10008_15_0_3_7
gdcm::UIDs, 736

uid_1_2_840_10008_15_0_3_8
gdcm::UIDs, 736

uid_1_2_840_10008_15_0_3_9
gdcm::UIDs, 736

uid_1_2_840_10008_15_0_4_1
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_4_2
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_4_3
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_4_4
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_4_5
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_4_6
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_4_7
gdcm::UIDs, 737

uid_1_2_840_10008_15_0_4_8
gdcm::UIDs, 737

uid_1_2_840_10008_1_1
gdcm::UIDs, 731

uid_1_2_840_10008_1_2
gdcm::UIDs, 731

uid_1_2_840_10008_1_20_1
gdcm::UIDs, 732

uid_1_2_840_10008_1_20_1_1
gdcm::UIDs, 732

uid_1_2_840_10008_1_20_2
gdcm::UIDs, 732

uid_1_2_840_10008_1_20_2_1
gdcm::UIDs, 732

uid_1_2_840_10008_1_2_1
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_1_99
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_2
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_100
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_50
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_51
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_52
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_53
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_54
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_55
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_56
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_57
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_58
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_59
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_60
gdcm::UIDs, 731

uid_1_2_840_10008_1_2_4_61
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_62
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_63
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_64
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_65
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_66
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_70
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_80
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_81
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_90
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_91
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_92
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_93
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_94
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_4_95
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_5
gdcml:UIDs, 731

uid_1_2_840_10008_1_2_6_1
gdcml:UIDs, 732

uid_1_2_840_10008_1_2_6_2
gdcml:UIDs, 732

uid_1_2_840_10008_1_3_10
gdcml:UIDs, 732

uid_1_2_840_10008_1_40
gdcml:UIDs, 732

uid_1_2_840_10008_1_40_1
gdcml:UIDs, 732

uid_1_2_840_10008_1_42
gdcml:UIDs, 732

uid_1_2_840_10008_1_42_1
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_1
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_10
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_11
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_12
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_13
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_14
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_15
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_16
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_17
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_18
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_2
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_3
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_4
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_5
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_6
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_7
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_8
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_1_9
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_2_1
gdcml:UIDs, 732

uid_1_2_840_10008_1_4_2_2
gdcml:UIDs, 732

uid_1_2_840_10008_1_9
gdcml:UIDs, 732

uid_1_2_840_10008_2_16_4
gdcml:UIDs, 732

uid_1_2_840_10008_2_6_1
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_1_1
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_2_1_1
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_2_1_4
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_2_2_1
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_2_3_1
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_2_3_2
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_2_3_3
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_2_3_4
gdcml:UIDs, 732

uid_1_2_840_10008_3_1_2_3_5
gdcm::UIDs, 733

uid_1_2_840_10008_3_1_2_5_1
gdcm::UIDs, 733

uid_1_2_840_10008_3_1_2_5_4
gdcm::UIDs, 733

uid_1_2_840_10008_3_1_2_5_5
gdcm::UIDs, 733

uid_1_2_840_10008_3_1_2_6_1
gdcm::UIDs, 733

uid_1_2_840_10008_4_2
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_14
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_15
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_16
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_16_376
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_17
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_17_376
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_18
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_18_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_2
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_22
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_23
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_24
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_24_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_25
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_26
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_27
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_29
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_30
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_31
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_32
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_33
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_4
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_4_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_4_2
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_9
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_1_9_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_10
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_104_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_104_2
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_11
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_11_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_11_2
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_11_3
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_11_4
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_128
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_129
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_12_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_12_1_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_12_2
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_12_2_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_12_3
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_13_1_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_13_1_2
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_1_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_1_1_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_1_2
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_1_2_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_1_3
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_1_3_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_2
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_20
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_2_1
gdcm::UIDs, 733

uid_1_2_840_10008_5_1_4_1_1_3
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_3_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_4
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_481_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_481_2
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_481_3
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_481_4
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_481_5
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_481_6
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_481_7
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_481_8
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_481_9
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_4_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_4_2
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_5
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_6
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_66
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_66_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_66_2
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_66_3
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_66_4
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_67
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_6_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_7
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_77_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_1_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_2
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_3
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_4
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_77_1_6
gdcm::UIDs, 737

uid_1_2_840_10008_5_1_4_1_1_77_2
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_7_1
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_7_2
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_7_3
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_7_4
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_8
gdcm::UIDs, 734

uid_1_2_840_10008_5_1_4_1_1_88_1
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_88_11
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_88_2
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_88_22
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_88_3
gdcm::UIDs, 735

uid_1_2_840_10008_5_1_4_1_1_88_33
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_1_88_4
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_1_88_40
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_1_88_50
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_1_88_59
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_1_88_65
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_1_88_67
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_1_9
 gdcm::UIDs, 734
 uid_1_2_840_10008_5_1_4_1_1_9_1
 gdcm::UIDs, 734
 uid_1_2_840_10008_5_1_4_1_1_9_1_1
 gdcm::UIDs, 734
 uid_1_2_840_10008_5_1_4_1_1_9_1_2
 gdcm::UIDs, 734
 uid_1_2_840_10008_5_1_4_1_1_9_1_3
 gdcm::UIDs, 734
 uid_1_2_840_10008_5_1_4_1_1_9_2_1
 gdcm::UIDs, 734
 uid_1_2_840_10008_5_1_4_1_1_9_3_1
 gdcm::UIDs, 734
 uid_1_2_840_10008_5_1_4_1_1_9_4_1
 gdcm::UIDs, 734
 uid_1_2_840_10008_5_1_4_1_2_1_1
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_2_1_2
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_2_1_3
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_2_2_1
 gdcm::UIDs, 735
 uid_1_2_840_10008_5_1_4_1_2_2_2
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_1_2_2_3
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_1_2_3_1
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_1_2_3_2
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_1_2_3_3
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_31
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_32
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_32_1
 gdcm::UIDs, 736

uid_1_2_840_10008_5_1_4_32_2
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_32_3
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_33
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_1
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_2
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_3
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_4
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_4_1
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_4_2
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_4_3
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_4_4
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_34_5
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_37_1
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_37_2
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_37_3
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_38_1
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_38_2
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_38_3
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_41
 gdcm::UIDs, 736
 uid_1_2_840_10008_5_1_4_42
 gdcm::UIDs, 736
 UltrasoundImageStorage
 gdcm::MediaStorage, 479
 gdcm::UIDs, 727
 UltrasoundImageStorageRetired
 gdcm::MediaStorage, 479
 gdcm::UIDs, 727
 UltrasoundMultiFrameImageStorage
 gdcm::MediaStorage, 479
 UltrasoundMultiFrameImageStorageRetired
 gdcm::MediaStorage, 479
 UltrasoundMultiframeImageStorage
 gdcm::UIDs, 727
 UltrasoundMultiframeImageStorageRetired
 gdcm::UIDs, 727

- UnInstallPipeline
 - vtkImageColorViewer, 841
- UnRegister
 - gdcm::Object, 507
- UndefinedEntityError
 - gdcm::Parser, 519
- underline
 - gdcm::terminal, 130
- UnexpectedStateError
 - gdcm::Parser, 519
- UnifiedProcedureStepEventSOPClass
 - gdcm::UIDs, 729
- UnifiedProcedureStepPullSOPClass
 - gdcm::UIDs, 729
- UnifiedProcedureStepPushSOPClass
 - gdcm::UIDs, 729
- UnifiedProcedureStepWatchSOPClass
 - gdcm::UIDs, 729
- UnifiedWorklistandProcedureStepSOPInstance
 - gdcm::UIDs, 729
- UnifiedWorklistandProcedureStepServiceClass
 - gdcm::UIDs, 729
- Unknown
 - gdcm::SwapCode, 684
 - gdcm::TransferSyntax, 710
- Unpack
 - gdcm::Unpacker12Bits, 790
- Update
 - gdcm::Curve, 265
 - gdcm::Overlay, 515
- UpdateDisplayExtent
 - vtkImageColorViewer, 841
- UpdateOrientation
 - vtkImageColorViewer, 841
- UpdatePosition
 - gdcm::ByteBuffer, 212
- Usage
 - gdcm::Usage, 791
- UsageType
 - gdcm::Usage, 791
- UseDictAlways
 - gdcm::PythonFilter, 574
 - gdcm::StringFilter, 667
- UserOption
 - gdcm::Usage, 791
- UserInfoormation
 - gdcm::network::UserInfoormation, 794
- UserOrdering
 - gdcm::SerieHelper, 627
- V
 - gdcm::Validate, 795
- VERBOSE_STYLE
 - gdcm::Printer, 564
- VERTEX
 - gdcm::MeshPrimitive, 489
- VIEWType_END
 - gdcm::Surface, 674
- VL16
 - gdcm::VR, 808
- VL32
 - gdcm::VR, 808
- VLEndoscopicImageStorage
 - gdcm::UIDs, 728
- VImageStorageTrialRetired
 - gdcm::UIDs, 728
- VLMicroscopicImageStorage
 - gdcm::UIDs, 728
- VMultiframeImageStorageTrialRetired
 - gdcm::UIDs, 728
- VPhotographicImageStorage
 - gdcm::MediaStorage, 481
 - gdcm::UIDs, 728
- VSlideCoordinatesMicroscopicImageStorage
 - gdcm::UIDs, 728
- VWholeSlideMicroscopyImageStorage
 - gdcm::MediaStorage, 481
 - gdcm::UIDs, 731
- VM0
 - gdcm::VM, 803
- VM1
 - gdcm::VM, 803
- VM10
 - gdcm::VM, 803
- VM12
 - gdcm::VM, 803
- VM16
 - gdcm::VM, 803
- VM18
 - gdcm::VM, 803
- VM1_2
 - gdcm::VM, 804
- VM1_3
 - gdcm::VM, 804
- VM1_32
 - gdcm::VM, 804
- VM1_4
 - gdcm::VM, 804
- VM1_5
 - gdcm::VM, 804
- VM1_8
 - gdcm::VM, 804
- VM1_99
 - gdcm::VM, 804
- VM1_n
 - gdcm::VM, 804
- VM2
 - gdcm::VM, 803

- VM24
 - gdcm::VM, 803
- VM256
 - gdcm::VM, 804
- VM28
 - gdcm::VM, 803
- VM2_2n
 - gdcm::VM, 804
- VM2_n
 - gdcm::VM, 804
- VM3
 - gdcm::VM, 803
- VM30_30n
 - gdcm::VM, 804
- VM32
 - gdcm::VM, 803
- VM35
 - gdcm::VM, 803
- VM3_3n
 - gdcm::VM, 804
- VM3_4
 - gdcm::VM, 804
- VM3_n
 - gdcm::VM, 804
- VM4
 - gdcm::VM, 803
- VM47_47n
 - gdcm::VM, 804
- VM4_4n
 - gdcm::VM, 804
- VM5
 - gdcm::VM, 803
- VM6
 - gdcm::VM, 803
- VM6_6n
 - gdcm::VM, 804
- VM7_7n
 - gdcm::VM, 804
- VM8
 - gdcm::VM, 803
- VM9
 - gdcm::VM, 803
- VM99
 - gdcm::VM, 804
- VM_END
 - gdcm::VM, 804
- VMType
 - gdcm::Attribute, 164
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 170
- VOILUTBoxSOPClass
 - gdcm::UIDs, 727
- VR_END
 - gdcm::VR, 808
- VR_VM1
 - gdcm::VR, 808
- VRALL
 - gdcm::VR, 808
- VRASCII
 - gdcm::VR, 808
- VRBINARY
 - gdcm::VR, 808
- VT100
 - gdcm::terminal, 130
- VIEWType
 - gdcm::Surface, 673
- VL
 - gdcm::VL, 800
- VM
 - gdcm::VM, 804
- VMType
 - gdcm::VM, 803
- VR
 - gdcm::VR, 808
- VRBINARY
 - gdcm, 123
- VRField
 - gdcm::CSAElement, 251
 - gdcm::DataElement, 274
- VRType
 - gdcm::VR, 807
- VRTypeTemplateCase
 - gdcmVR.h, 945
- VTK_CMYK
 - vtkGDCMImageReader.h, 947
- VTK_LEGACY
 - vtkImageColorViewer, 841
- VTK_LOOKUP_TABLE
 - vtkGDCMImageReader.h, 947
- VTK_YBR
 - vtkGDCMImageReader.h, 947
- Valid
 - gdcm::Preamble, 553
- Validate
 - gdcm::PixelFormat, 537
 - gdcm::Validate, 795
- ValidateQuery
 - gdcm::BaseRootQuery, 195
 - gdcm::FindPatientRootQuery, 371
 - gdcm::FindStudyRootQuery, 373
 - gdcm::MovePatientRootQuery, 499
 - gdcm::MoveStudyRootQuery, 501
- Validation
 - gdcm::Validate, 795
- Value
 - gdcm::Value, 797
- value
 - gdcm::SerieHelper::Rule, 597

- gdcmm::STATIC_ASSERTION_FAILURE< true >, 652
- value_type
 - gdcmm::CodeString, 233
 - gdcmm::LO, 466
 - gdcmm::String, 664
- ValueField
 - gdcmm::DataElement, 274
 - gdcmm::PDSElement, 524
- ValueLengthField
 - gdcmm::DataElement, 274
- ValueMultiplicityField
 - gdcmm::CSAElement, 251
- ValuePtr
 - gdcmm::DataElement, 268
- ValueType
 - gdcmm::Scanner, 601
- VerificationSOPClass
 - gdcmm::UIDs, 724
- Verify
 - gdcmm::Defs, 291
 - gdcmm::Macro, 473
 - gdcmm::Module, 493
- Version
 - gdcmm::Version, 798
- Video
 - gdcmm::MediaStorage, 481
- VideoEndoscopicImageStorage
 - gdcmm::MediaStorage, 480
 - gdcmm::UIDs, 728
- VideoMicroscopicImageStorage
 - gdcmm::UIDs, 728
- VideoPhotographicImageStorage
 - gdcmm::UIDs, 728
- vtkGDCMImageWriter
 - JPEG2000_COMPRESSION, 820
 - JPEG_COMPRESSION, 820
 - JPEGLS_COMPRESSION, 820
 - NO_COMPRESSION, 820
 - RLE_COMPRESSION, 820
- vtkImageColorViewer
 - SLICE_ORIENTATION_XY, 838
 - SLICE_ORIENTATION_XZ, 838
 - SLICE_ORIENTATION_YZ, 838
- vtkBooleanMacro
 - vtkGDCMImageReader, 816, 817
 - vtkGDCMImageWriter, 821
 - vtkGDCMThreadedImageReader, 833
 - vtkGDCMThreadedImageReader2, 835
 - vtkImageColorViewer, 841
 - vtkImageMapToColors16, 843
- vtkGDCMImageReader, 813
 - ~vtkGDCMImageReader, 815
 - ApplyInverseVideo, 818
 - ApplyLookupTable, 818
 - ApplyPlanarConfiguration, 818
 - ApplyShiftScale, 818
 - ApplyYBRToRGB, 818
 - CanReadFile, 815
 - Curve, 818
 - DirectionCosines, 818
 - ExecuteData, 815
 - ExecuteInformation, 815
 - FileNames, 818
 - FillMedicalImageInformation, 815
 - ForceRescale, 818
 - GetDescriptiveName, 816
 - GetFileExtensions, 816
 - GetIconImage, 816
 - GetOverlay, 816
 - IconDataScalarType, 818
 - IconImageDataExtent, 818
 - IconNumberOfScalarComponents, 818
 - ImageFormat, 818
 - ImageOrientationPatient, 818
 - ImagePositionPatient, 818
 - LoadIconImage, 818
 - LoadOverlays, 818
 - LoadSingleFile, 816
 - LossyFlag, 818
 - MedicalImageProperties, 818
 - New, 816
 - NumberOfIconImages, 818
 - NumberOfOverlays, 818
 - PlanarConfiguration, 818
 - PrintSelf, 816
 - RequestDataCompat, 816
 - RequestInformationCompat, 816
 - Scale, 819
 - SetCurve, 816
 - SetFileNames, 816
 - SetFilePattern, 816
 - SetFilePrefix, 816
 - SetMedicalImageProperties, 816
 - Shift, 819
 - vtkBooleanMacro, 816, 817
 - vtkGDCMImageReader, 815
 - vtkGetMacro, 817
 - vtkGetObjectMacro, 817
 - vtkGetStringMacro, 817
 - vtkGetVector3Macro, 817
 - vtkGetVector6Macro, 817
 - vtkSetMacro, 817, 818
 - vtkSetVector6Macro, 818
 - vtkTypeRevisionMacro, 818
 - vtkGDCMImageReader, 815
 - vtkGDCMMedicalImageProperties, 824
 - vtkGDCMImageReader.h, 947

- VTK_CMYK, 947
- VTK_YBR, 947
- vtkGDCMImageWriter, 819
 - ~vtkGDCMImageWriter, 820
 - CompressionTypes, 820
 - GetDescriptiveName, 820
 - GetFileExtensions, 821
 - GetFileName, 821
 - New, 821
 - PrintSelf, 821
 - SetDirectionCosines, 821
 - SetDirectionCosinesFromImageOrientationPatient, 821
 - SetFileNames, 821
 - SetMedicalImageProperties, 821
 - vtkBooleanMacro, 821
 - vtkGDCMImageWriter, 820
 - vtkGetMacro, 821, 822
 - vtkGetObjectMacro, 822
 - vtkGetStringMacro, 822
 - vtkSetMacro, 822
 - vtkSetStringMacro, 822
 - vtkTypeRevisionMacro, 822
 - vtkGDCMImageWriter, 820
 - vtkGDCMMedicalImageProperties, 824
 - Write, 822
 - WriteGDCMData, 822
 - WriteSlice, 822
- vtkGDCMImageWriter.h, 947
- vtkGDCMMedicalImageProperties, 823
 - ~vtkGDCMMedicalImageProperties, 823
 - Clear, 823
 - GetFile, 823
 - New, 823
 - PrintSelf, 823
 - PushBackFile, 823
 - vtkGDCMImageReader, 824
 - vtkGDCMImageWriter, 824
 - vtkGDCMMedicalImageProperties, 823
 - vtkTypeRevisionMacro, 823
 - vtkGDCMMedicalImageProperties, 823
- vtkGDCMMedicalImageProperties.h, 948
- vtkGDCMPolyDataReader, 824
 - ~vtkGDCMPolyDataReader, 825
 - FileName, 826
 - FillMedicalImageInformation, 825
 - MedicalImageProperties, 826
 - New, 825
 - PrintSelf, 825
 - RTStructSetProperties, 826
 - RequestData, 825
 - RequestData_HemodynamicWaveformStorage, 825
 - RequestData_RTStructureSetStorage, 825
 - RequestInformation, 826
 - RequestInformation_HemodynamicWaveformStorage, 826
 - RequestInformation_RTStructureSetStorage, 826
- vtkGDCMPolyDataReader, 825
 - vtkGetObjectMacro, 826
 - vtkGetStringMacro, 826
 - vtkSetStringMacro, 826
 - vtkTypeRevisionMacro, 826
 - vtkGDCMPolyDataReader, 825
- vtkGDCMPolyDataReader.h, 948
- vtkGDCMPolyDataWriter, 826
 - ~vtkGDCMPolyDataWriter, 828
 - InitializeRTStructSet, 828
 - MedicalImageProperties, 829
 - New, 828
 - PrintSelf, 828
 - RTStructSetProperties, 829
 - SetMedicalImageProperties, 828
 - SetNumberOfInputPorts, 828
 - SetRTStructSetProperties, 828
 - vtkGDCMPolyDataWriter, 828
 - vtkTypeRevisionMacro, 828
 - vtkGDCMPolyDataWriter, 828
 - WriteData, 828
 - WriteRTSTRUCTData, 828
 - WriteRTSTRUCTInfo, 828
- vtkGDCMPolyDataWriter.h, 948
- vtkGDCMTesting, 829
 - ~vtkGDCMTesting, 830
 - GetGDCMDataRoot, 830
 - GetMD5MetaImage, 830
 - GetMHDMD5FromFile, 830
 - GetNumberOfMD5MetaImages, 830
 - GetRAWMD5FromFile, 830
 - GetVTKDataRoot, 830
 - MD5MetaImagesType, 830
 - New, 830
 - PrintSelf, 830
 - vtkGDCMTesting, 830
 - vtkTypeRevisionMacro, 831
 - vtkGDCMTesting, 830
- vtkGDCMTesting.h, 948
- vtkGDCMThreadedImageReader, 831
 - ~vtkGDCMThreadedImageReader, 832
 - ExecuteData, 832
 - ExecuteInformation, 832
 - New, 832
 - PrintSelf, 832
 - ReadFiles, 832
 - RequestDataCompat, 833
 - vtkBooleanMacro, 833
 - vtkGDCMThreadedImageReader, 832
 - vtkGetMacro, 833
 - vtkSetMacro, 833

- vtkTypeRevisionMacro, 833
- vtkGDCMThreadedImageReader, 832
- vtkGDCMThreadedImageReader.h, 949
- vtkGDCMThreadedImageReader2, 833
 - ~vtkGDCMThreadedImageReader2, 834
 - GetFileName, 834
 - New, 834
 - PrintSelf, 834
 - RequestInformation, 834
 - SetFileName, 834
 - SetFileNames, 834
 - SplitExtent, 834
 - ThreadedRequestData, 834
- vtkBooleanMacro, 835
- vtkGDCMThreadedImageReader2, 834
- vtkGetMacro, 835
- vtkGetObjectMacro, 835
- vtkGetVector3Macro, 835
- vtkGetVector6Macro, 835
- vtkSetMacro, 835
- vtkSetVector3Macro, 835
- vtkSetVector6Macro, 835
- vtkTypeRevisionMacro, 836
- vtkGDCMThreadedImageReader2, 834
- vtkGDCMThreadedImageReader2.h, 949
- vtkGetMacro
 - vtkGDCMImageReader, 817
 - vtkGDCMImageWriter, 821, 822
 - vtkGDCMThreadedImageReader, 833
 - vtkGDCMThreadedImageReader2, 835
 - vtkImageColorViewer, 841
 - vtkImageMapToColors16, 844
 - vtkImageMapToWindowLevelColors2, 845
- vtkGetObjectMacro
 - vtkGDCMImageReader, 817
 - vtkGDCMImageWriter, 822
 - vtkGDCMPolyDataReader, 826
 - vtkGDCMThreadedImageReader2, 835
 - vtkImageColorViewer, 841
 - vtkImageMapToColors16, 844
- vtkGetStringMacro
 - vtkGDCMImageReader, 817
 - vtkGDCMImageWriter, 822
 - vtkGDCMPolyDataReader, 826
 - vtkRTStructSetProperties, 852, 853
- vtkGetVector3Macro
 - vtkGDCMImageReader, 817
 - vtkGDCMThreadedImageReader2, 835
- vtkGetVector6Macro
 - vtkGDCMImageReader, 817
 - vtkGDCMThreadedImageReader2, 835
- vtkImageColorViewer, 836
 - ~vtkImageColorViewer, 838
 - AddInput, 838
 - AddInputConnection, 838
 - FirstRender, 841
 - GetColorLevel, 838
 - GetColorWindow, 838
 - GetInput, 839
 - GetOffScreenRendering, 839
 - GetOverlayVisibility, 839
 - GetPosition, 839
 - GetSize, 839
 - GetSliceMax, 839
 - GetSliceMin, 839
 - GetSliceRange, 839
 - GetWindowName, 839
 - ImageActor, 841
 - InstallPipeline, 839
 - Interactor, 841
 - InteractorStyle, 842
 - New, 839
 - OverlayImageActor, 842
 - PrintSelf, 839
 - Render, 839
 - RenderWindow, 842
 - Renderer, 842
 - SetColorLevel, 839
 - SetColorWindow, 839
 - SetDisplayId, 839
 - SetInput, 839
 - SetInputConnection, 840
 - SetOffScreenRendering, 840
 - SetOverlayVisibility, 840
 - SetParentId, 840
 - SetPosition, 840
 - SetRenderWindow, 840
 - SetRenderer, 840
 - SetSize, 840
 - SetSlice, 840
 - SetSliceOrientation, 840
 - SetSliceOrientationToXY, 840
 - SetSliceOrientationToXZ, 840
 - SetSliceOrientationToYZ, 840
 - SetWindowId, 841
 - SetupInteractor, 841
 - Slice, 842
 - SliceOrientation, 842
 - UnInstallPipeline, 841
 - UpdateDisplayExtent, 841
 - UpdateOrientation, 841
 - VTK_LEGACY, 841
 - vtkBooleanMacro, 841
 - vtkGetMacro, 841
 - vtkGetObjectMacro, 841
 - vtkImageColorViewer, 838
 - vtkTypeRevisionMacro, 841
 - vtkImageColorViewer, 838

- WindowLevel, 842
- vtkImageColorViewer.h, 949
- vtkImageMapToColors16, 842
 - ~vtkImageMapToColors16, 843
 - ActiveComponent, 844
 - DataWasPassed, 844
 - GetMTime, 843
 - LookupTable, 844
 - New, 843
 - OutputFormat, 844
 - PassAlphaToOutput, 844
 - PrintSelf, 843
 - RequestData, 843
 - RequestInformation, 843
 - SetLookupTable, 843
 - SetOutputFormatToLuminance, 843
 - SetOutputFormatToLuminanceAlpha, 843
 - SetOutputFormatToRGB, 843
 - SetOutputFormatToRGBA, 843
 - ThreadedRequestData, 843
 - vtkBooleanMacro, 843
 - vtkGetMacro, 844
 - vtkGetObjectMacro, 844
 - vtkImageMapToColors16, 843
 - vtkSetMacro, 844
 - vtkTypeRevisionMacro, 844
 - vtkImageMapToColors16, 843
- vtkImageMapToColors16.h, 949
- vtkImageMapToWindowLevelColors2, 844
 - ~vtkImageMapToWindowLevelColors2, 845
 - Level, 846
 - New, 845
 - PrintSelf, 845
 - RequestData, 845
 - RequestInformation, 845
 - ThreadedRequestData, 845
 - vtkGetMacro, 845
 - vtkImageMapToWindowLevelColors2, 845
 - vtkSetMacro, 845
 - vtkTypeRevisionMacro, 845
 - vtkImageMapToWindowLevelColors2, 845
 - Window, 846
- vtkImageMapToWindowLevelColors2.h, 949
- vtkImagePlanarComponentsToComponents, 846
 - ~vtkImagePlanarComponentsToComponents, 846
 - New, 846
 - PrintSelf, 846
 - RequestData, 846
 - vtkImagePlanarComponentsToComponents, 846
 - vtkTypeRevisionMacro, 847
 - vtkImagePlanarComponentsToComponents, 846
- vtkImagePlanarComponentsToComponents.h, 949
- vtkImageRGBToYBR, 847
 - ~vtkImageRGBToYBR, 847

- New, 847
- PrintSelf, 847
- ThreadedExecute, 847
- vtkImageRGBToYBR, 847
- vtkTypeRevisionMacro, 847
- vtkImageRGBToYBR, 847
- vtkImageRGBToYBR.h, 949
- vtkImageYBRToRGB, 848
 - ~vtkImageYBRToRGB, 848
 - New, 848
 - PrintSelf, 848
 - ThreadedExecute, 848
 - vtkImageYBRToRGB, 848
 - vtkTypeRevisionMacro, 848
 - vtkImageYBRToRGB, 848
- vtkImageYBRToRGB.h, 950
- vtkLookupTable16, 848
 - ~vtkLookupTable16, 849
 - Build, 849
 - GetPointer, 849
 - MapScalarsThroughTable2, 849
 - New, 849
 - PrintSelf, 849
 - SetNumberOfTableValues, 849
 - Table16, 850
 - vtkLookupTable16, 849
 - vtkTypeRevisionMacro, 849
 - vtkLookupTable16, 849
 - WritePointer, 849
- vtkLookupTable16.h, 950
- vtkRTStructSetProperties, 850
 - ~vtkRTStructSetProperties, 851
 - AddContourReferencedFrameOfReference, 851
 - AddReferencedFrameOfReference, 851
 - AddStructureSetROI, 851
 - AddStructureSetROIObservation, 852
 - Clear, 852
 - DeepCopy, 852
 - GetContourReferencedFrameOfReferenceClassUID, 852
 - GetContourReferencedFrameOfReferenceInstanceUID, 852
 - GetNumberOfContourReferencedFrameOfReferences, 852
 - GetNumberOfReferencedFrameOfReferences, 852
 - GetNumberOfStructureSetROIs, 852
 - GetReferencedFrameOfReferenceClassUID, 852
 - GetReferencedFrameOfReferenceInstanceUID, 852
 - GetStructureSetObservationNumber, 852
 - GetStructureSetROIGenerationAlgorithm, 852
 - GetStructureSetROIName, 852
 - GetStructureSetROIName, 852
 - GetStructureSetROIRefFrameRefUID, 852
 - GetStructureSetRTROIInterpretedType, 852

- Internals, 853
- New, 852
- PrintSelf, 852
- ReferenceFrameOfReferenceUID, 853
- ReferenceSeriesInstanceUID, 853
- SOPInstanceUID, 853
- SeriesInstanceUID, 853
- StructureSetDate, 853
- StructureSetLabel, 853
- StructureSetName, 853
- StructureSetTime, 853
- StudyInstanceUID, 854
- vtkGetStringMacro, 852, 853
- vtkRTStructSetProperties, 851
- vtkSetStringMacro, 853
- vtkTypeRevisionMacro, 853
- vtkRTStructSetProperties, 851
- vtkRTStructSetProperties.h, 950
- vtkSetMacro
 - vtkGDCMImageReader, 817, 818
 - vtkGDCMImageWriter, 822
 - vtkGDCMThreadedImageReader, 833
 - vtkGDCMThreadedImageReader2, 835
 - vtkImageMapToColors16, 844
 - vtkImageMapToWindowLevelColors2, 845
- vtkSetStringMacro
 - vtkGDCMImageWriter, 822
 - vtkGDCMPolyDataReader, 826
 - vtkRTStructSetProperties, 853
- vtkSetVector3Macro
 - vtkGDCMThreadedImageReader2, 835
- vtkSetVector6Macro
 - vtkGDCMImageReader, 818
 - vtkGDCMThreadedImageReader2, 835
- vtkTypeRevisionMacro
 - vtkGDCMImageReader, 818
 - vtkGDCMImageWriter, 822
 - vtkGDCMMedicalImageProperties, 823
 - vtkGDCMPolyDataReader, 826
 - vtkGDCMPolyDataWriter, 828
 - vtkGDCMTesting, 831
 - vtkGDCMThreadedImageReader, 833
 - vtkGDCMThreadedImageReader2, 836
 - vtkImageColorViewer, 841
 - vtkImageMapToColors16, 844
 - vtkImageMapToWindowLevelColors2, 845
 - vtkImagePlanarComponentsToComponents, 847
 - vtkImageRGBToYBR, 847
 - vtkImageYBRToRGB, 848
 - vtkLookupTable16, 849
 - vtkRTStructSetProperties, 853
- WarningOff
 - gdcm::Trace, 708
- WarningOn
 - gdcm::Trace, 708
- Waveform
 - gdcm::Waveform, 854
 - gdcm::MediaStorage, 481
- WaveformStorageTrialRetired
 - gdcm::UIDs, 727
- what
 - gdcm::Exception, 343
- white
 - gdcm::terminal, 130
- Window
 - vtkImageMapToWindowLevelColors2, 846
- WindowLevel
 - vtkImageColorViewer, 842
- Write
 - gdcm::ByteValue, 218
 - gdcm::CommandDataSet, 238
 - gdcm::CSAHeader, 255
 - gdcm::DataElement, 274
 - gdcm::DataSet, 284
 - gdcm::Element, 320
 - gdcm::Element< TVR, VM::VM1_n >, 325
 - gdcm::EncodingImplementation< VR::VRASCII >, 337
 - gdcm::EncodingImplementation< VR::VRBINARY >, 338
 - gdcm::ExplicitDataElement, 346
 - gdcm::File, 352
 - gdcm::FileMetaInformation, 361
 - gdcm::Fragment, 376
 - gdcm::ImageWriter, 429
 - gdcm::ImplicitDataElement, 433
 - gdcm::Item, 446
 - gdcm::network::AAAbortPDU, 135
 - gdcm::network::AAssociateACPDU, 137
 - gdcm::network::AAssociateRJPDU, 139
 - gdcm::network::AAssociateRQPDU, 142
 - gdcm::network::AbstractSyntax, 144
 - gdcm::network::ApplicationContext, 154
 - gdcm::network::AReleaseRPPDU, 157
 - gdcm::network::AReleaseRQPDU, 159
 - gdcm::network::AsynchronousOperationsWindow-Sub, 162
 - gdcm::network::BasePDU, 193
 - gdcm::network::ImplementationClassUIDSub, 430
 - gdcm::network::ImplementationUIDSub, 431
 - gdcm::network::ImplementationVersionNameSub, 431
 - gdcm::network::MaximumLengthSub, 475
 - gdcm::network::PDataTFPDU, 522
 - gdcm::network::PresentationContextAC, 556
- WIREFRAME
 - gdcm::Surface, 674

- gdcm::network::PresentationContextRQ, 560
- gdcm::network::PresentationDataValue, 562
- gdcm::network::TransferSyntaxSub, 713
- gdcm::network::UserInformation, 794
- gdcm::PixmapWriter, 549
- gdcm::PNMCodec, 551
- gdcm::Preamble, 553
- gdcm::SegmentWriter, 614
- gdcm::SequenceOfFragments, 618
- gdcm::SequenceOfItems, 624
- gdcm::StreamImageWriter, 660
- gdcm::SurfaceWriter, 683
- gdcm::Tag, 701
- gdcm::ValueIO, 798
- gdcm::VL, 801
- gdcm::VR, 809
- gdcm::VRVLSize< 0 >, 812
- gdcm::VRVLSize< 1 >, 813
- gdcm::Writer, 859
- itk::GDCMImageIO2, 383
- vtkGDCMImageWriter, 822
- Write16
 - gdcm::VL, 801
- WriteASCII
 - gdcm::Element< TVR, VM::VM1_n >, 325
- WriteBuffer
 - gdcm::ByteValue, 218
 - gdcm::SequenceOfFragments, 618
- WriteBufferAsRGBA
 - gdcm::LookupTable, 471
- WriteData
 - vtkGDCMPolyDataWriter, 828
- WriteFooter
 - gdcm::DictConverter, 300
- WriteGDCMData
 - vtkGDCMImageWriter, 822
- WriteHeader
 - gdcm::DictConverter, 301
- WriteHelpFile
 - gdcm::BaseRootQuery, 196
- WriteImageInformation
 - gdcm::StreamImageWriter, 661
 - itk::GDCMImageIO2, 383
- WriteImageSubregionRAW
 - gdcm::StreamImageWriter, 661
- WritePointer
 - vtkLookupTable16, 849
- WriteQuery
 - gdcm::BaseRootQuery, 196
- WriteRTSTRUCTData
 - vtkGDCMPolyDataWriter, 828
- WriteRTSTRUCTInfo
 - vtkGDCMPolyDataWriter, 828
- WriteRawHeader
 - gdcm::StreamImageWriter, 661
- WriteSlice
 - vtkGDCMImageWriter, 822
- Writer
 - gdcm::Writer, 858
- XML
 - gdcm::Printer, 564
- XMLEncoding
 - gdcm::UIDs, 725
- XR3DAngiographicImageStorage
 - gdcm::MediaStorage, 481
 - gdcm::UIDs, 728
- XR3DCraniofacialImageStorage
 - gdcm::UIDs, 728
- XR3DAngiographicBiPlaneImageStorageRetired
 - gdcm::MediaStorage, 480
 - gdcm::UIDs, 728
- XR3DAngiographicImageStorage
 - gdcm::MediaStorage, 480
 - gdcm::UIDs, 728
- XR3DRadiationDoseSRStorage
 - gdcm::UIDs, 729
- XR3DRadiofluoroscopicImageStorage
 - gdcm::UIDs, 728
- XR3DRadiofluoroscopicImageStorage
 - gdcm::MediaStorage, 480
- XMLDictReader
 - gdcm::XMLDictReader, 861
- XMLPrivateDictReader
 - gdcm::XMLPrivateDictReader, 863
- YBR_FULL
 - gdcm::PhotometricInterpretation, 532
- YBR_FULL_422
 - gdcm::PhotometricInterpretation, 532
- YBR_ICT
 - gdcm::PhotometricInterpretation, 532
- YBR_PARTIAL_420
 - gdcm::PhotometricInterpretation, 532
- YBR_PARTIAL_422
 - gdcm::PhotometricInterpretation, 532
- YBR_RCT
 - gdcm::PhotometricInterpretation, 532
- YES
 - gdcm::Surface, 673
- YBR2RGB
 - gdcm::ImageChangePhotometricInterpretation, 402
- yellow
 - gdcm::terminal, 130
- ZEROED_OUT
 - gdcm::CSAHeader, 253
- ZSpacing
 - gdcm::IPPSorter, 443

ZTolerance
gdc::IPPSorter, 443