

Spinnaker C++

1.18.0.17

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Software Licensing Information</b>	<b>3</b>
<b>3</b>	<b>Module Index</b>	<b>5</b>
3.1	Modules . . . . .	5
<b>4</b>	<b>Namespace Index</b>	<b>9</b>
4.1	Namespace List . . . . .	9
<b>5</b>	<b>Hierarchical Index</b>	<b>11</b>
5.1	Class Hierarchy . . . . .	11
<b>6</b>	<b>Class Index</b>	<b>17</b>
6.1	Class List . . . . .	17
<b>7</b>	<b>File Index</b>	<b>23</b>
7.1	File List . . . . .	23
<b>8</b>	<b>Module Documentation</b>	<b>27</b>
8.1	Spinnaker Event Classes . . . . .	27
8.1.1	Detailed Description . . . . .	28
8.2	ArrivalEvent Class . . . . .	29
8.2.1	Detailed Description . . . . .	29
8.3	Spinnaker Classes . . . . .	30
8.3.1	Detailed Description . . . . .	32
8.4	AVI Recorder Class . . . . .	33

8.4.1	Detailed Description	33
8.4.2	Function Documentation	33
8.4.2.1	DEPRECATED_CLASS("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API AVIRecorder	33
8.5	BasePtr Class	35
8.5.1	Detailed Description	35
8.6	Camera Class	36
8.6.1	Detailed Description	36
8.7	Camera Base Class	37
8.7.1	Detailed Description	37
8.8	CameraDefs Class	38
8.8.1	Detailed Description	68
8.8.2	Enumeration Type Documentation	68
8.8.2.1	AcquisitionModeEnums	68
8.8.2.2	AcquisitionStatusSelectorEnums	69
8.8.2.3	ActionUnconditionalModeEnums	69
8.8.2.4	AdcBitDepthEnums	69
8.8.2.5	AutoAlgorithmSelectorEnums	69
8.8.2.6	AutoExposureControlPriorityEnums	70
8.8.2.7	AutoExposureLightingModeEnums	70
8.8.2.8	AutoExposureMeteringModeEnums	70
8.8.2.9	AutoExposureTargetGreyValueAutoEnums	71
8.8.2.10	BalanceRatioSelectorEnums	71
8.8.2.11	BalanceWhiteAutoEnums	71
8.8.2.12	BalanceWhiteAutoProfileEnums	71
8.8.2.13	BinningHorizontalModeEnums	72
8.8.2.14	BinningSelectorEnums	72
8.8.2.15	BinningVerticalModeEnums	72
8.8.2.16	BlackLevelAutoBalanceEnums	72
8.8.2.17	BlackLevelAutoEnums	73
8.8.2.18	BlackLevelSelectorEnums	73



8.8.2.19	ChunkBlackLevelSelectorEnums	73
8.8.2.20	ChunkCounterSelectorEnums	73
8.8.2.21	ChunkEncoderSelectorEnums	74
8.8.2.22	ChunkEncoderStatusEnums	74
8.8.2.23	ChunkExposureTimeSelectorEnums	74
8.8.2.24	ChunkGainSelectorEnums	75
8.8.2.25	ChunkImageComponentEnums	75
8.8.2.26	ChunkPixelFormatEnums	75
8.8.2.27	ChunkRegionIDEnums	76
8.8.2.28	ChunkScan3dCoordinateReferenceSelectorEnums	76
8.8.2.29	ChunkScan3dCoordinateSelectorEnums	76
8.8.2.30	ChunkScan3dCoordinateSystemEnums	76
8.8.2.31	ChunkScan3dCoordinateSystemReferenceEnums	77
8.8.2.32	ChunkScan3dCoordinateTransformSelectorEnums	77
8.8.2.33	ChunkScan3dDistanceUnitEnums	77
8.8.2.34	ChunkScan3dOutputModeEnums	78
8.8.2.35	ChunkSelectorEnums	78
8.8.2.36	ChunkSourceIDEnums	79
8.8.2.37	ChunkTimerSelectorEnums	79
8.8.2.38	ChunkTransferStreamIDEnums	79
8.8.2.39	CICongfigurationEnums	80
8.8.2.40	CITimeSlotsCountEnums	80
8.8.2.41	ColorTransformationSelectorEnums	80
8.8.2.42	ColorTransformationValueSelectorEnums	81
8.8.2.43	CounterEventActivationEnums	81
8.8.2.44	CounterEventSourceEnums	81
8.8.2.45	CounterResetActivationEnums	82
8.8.2.46	CounterResetSourceEnums	82
8.8.2.47	CounterSelectorEnums	83
8.8.2.48	CounterStatusEnums	83

8.8.2.49	CounterTriggerActivationEnums . . . . .	83
8.8.2.50	CounterTriggerSourceEnums . . . . .	84
8.8.2.51	CxpConnectionTestModeEnums . . . . .	84
8.8.2.52	CxpLinkConfigurationEnums . . . . .	84
8.8.2.53	CxpLinkConfigurationPreferredEnums . . . . .	85
8.8.2.54	CxpLinkConfigurationStatusEnums . . . . .	86
8.8.2.55	CxpPoCxpStatusEnums . . . . .	87
8.8.2.56	DecimationHorizontalModeEnums . . . . .	87
8.8.2.57	DecimationSelectorEnums . . . . .	88
8.8.2.58	DecimationVerticalModeEnums . . . . .	88
8.8.2.59	DefectCorrectionModeEnums . . . . .	88
8.8.2.60	DeinterlacingEnums . . . . .	88
8.8.2.61	DeviceCharacterSetEnums . . . . .	89
8.8.2.62	DeviceClockSelectorEnums . . . . .	89
8.8.2.63	DeviceConnectionStatusEnums . . . . .	89
8.8.2.64	DeviceIndicatorModeEnums . . . . .	89
8.8.2.65	DeviceLinkHeartbeatModeEnums . . . . .	90
8.8.2.66	DeviceLinkThroughputLimitModeEnums . . . . .	90
8.8.2.67	DevicePowerSupplySelectorEnums . . . . .	90
8.8.2.68	DeviceRegistersEndiannessEnums . . . . .	90
8.8.2.69	DeviceScanTypeEnums . . . . .	90
8.8.2.70	DeviceSerialPortBaudRateEnums . . . . .	91
8.8.2.71	DeviceSerialPortSelectorEnums . . . . .	91
8.8.2.72	DeviceStreamChannelEndiannessEnums . . . . .	91
8.8.2.73	DeviceStreamChannelTypeEnums . . . . .	91
8.8.2.74	DeviceTapGeometryEnums . . . . .	92
8.8.2.75	DeviceTemperatureSelectorEnums . . . . .	93
8.8.2.76	DeviceTLTypeEnums . . . . .	93
8.8.2.77	DeviceTypeEnums . . . . .	93
8.8.2.78	EncoderModeEnums . . . . .	94

8.8.2.79	EncoderOutputModeEnums	94
8.8.2.80	EncoderResetActivationEnums	94
8.8.2.81	EncoderResetSourceEnums	95
8.8.2.82	EncoderSelectorEnums	96
8.8.2.83	EncoderSourceAEnums	96
8.8.2.84	EncoderSourceBEnums	96
8.8.2.85	EncoderStatusEnums	96
8.8.2.86	EventNotificationEnums	97
8.8.2.87	EventSelectorEnums	97
8.8.2.88	ExposureActiveModeEnums	97
8.8.2.89	ExposureAutoEnums	97
8.8.2.90	ExposureModeEnums	98
8.8.2.91	ExposureTimeModeEnums	98
8.8.2.92	ExposureTimeSelectorEnums	98
8.8.2.93	FileOpenModeEnums	99
8.8.2.94	FileOperationSelectorEnums	99
8.8.2.95	FileOperationStatusEnums	99
8.8.2.96	FileSelectorEnums	99
8.8.2.97	GainAutoBalanceEnums	100
8.8.2.98	GainAutoEnums	100
8.8.2.99	GainSelectorEnums	100
8.8.2.100	GevCCPEnums	100
8.8.2.101	GevCurrentPhysicalLinkConfigurationEnums	101
8.8.2.102	GevGVCPExtendedStatusCodesSelectorEnums	101
8.8.2.103	GevGVSPExtendedIDModeEnums	101
8.8.2.104	GevIEEE1588ClockAccuracyEnums	101
8.8.2.105	GevIEEE1588ModeEnums	102
8.8.2.106	GevIEEE1588StatusEnums	102
8.8.2.107	GevIPConfigurationStatusEnums	102
8.8.2.108	GevPhysicalLinkConfigurationEnums	102

8.8.2.109	GevSupportedOptionSelectorEnums	103
8.8.2.110	ImageComponentSelectorEnums	103
8.8.2.111	ImageCompressionJPEGFormatOptionEnums	104
8.8.2.112	ImageCompressionModeEnums	104
8.8.2.113	ImageCompressionRateOptionEnums	104
8.8.2.114	LineFormatEnums	105
8.8.2.115	LineInputFilterSelectorEnums	105
8.8.2.116	LineModeEnums	105
8.8.2.117	LineSelectorEnums	105
8.8.2.118	LineSourceEnums	106
8.8.2.119	LogicBlockLUTInputActivationEnums	106
8.8.2.120	LogicBlockLUTInputSelectorEnums	106
8.8.2.121	LogicBlockLUTInputSourceEnums	107
8.8.2.122	LogicBlockLUTSelectorEnums	107
8.8.2.123	LogicBlockSelectorEnums	107
8.8.2.124	LUTSelectorEnums	108
8.8.2.125	PixelColorFilterEnums	108
8.8.2.126	PixelFormatEnums	108
8.8.2.127	PixelFormatInfoSelectorEnums	114
8.8.2.128	PixelSizeEnums	119
8.8.2.129	RegionDestinationEnums	119
8.8.2.130	RegionModeEnums	120
8.8.2.131	RegionSelectorEnums	120
8.8.2.132	RgbTransformLightSourceEnums	120
8.8.2.133	Scan3dCoordinateReferenceSelectorEnums	121
8.8.2.134	Scan3dCoordinateSelectorEnums	121
8.8.2.135	Scan3dCoordinateSystemEnums	121
8.8.2.136	Scan3dCoordinateSystemReferenceEnums	121
8.8.2.137	Scan3dCoordinateTransformSelectorEnums	122
8.8.2.138	Scan3dDistanceUnitEnums	122

8.8.2.139 Scan3dOutputModeEnums . . . . .	122
8.8.2.140 SensorDigitizationTapsEnums . . . . .	123
8.8.2.141 SensorShutterModeEnums . . . . .	123
8.8.2.142 SensorTapsEnums . . . . .	123
8.8.2.143 SequencerConfigurationModeEnums . . . . .	124
8.8.2.144 SequencerConfigurationValidEnums . . . . .	124
8.8.2.145 SequencerModeEnums . . . . .	124
8.8.2.146 SequencerSetValidEnums . . . . .	124
8.8.2.147 SequencerTriggerActivationEnums . . . . .	124
8.8.2.148 SequencerTriggerSourceEnums . . . . .	125
8.8.2.149 SerialPortBaudRateEnums . . . . .	125
8.8.2.150 SerialPortParityEnums . . . . .	125
8.8.2.151 SerialPortSelectorEnums . . . . .	126
8.8.2.152 SerialPortSourceEnums . . . . .	126
8.8.2.153 SerialPortStopBitsEnums . . . . .	126
8.8.2.154 SoftwareSignalSelectorEnums . . . . .	126
8.8.2.155 SourceSelectorEnums . . . . .	127
8.8.2.156 TestPatternEnums . . . . .	127
8.8.2.157 TestPatternGeneratorSelectorEnums . . . . .	127
8.8.2.158 TimerSelectorEnums . . . . .	127
8.8.2.159 TimerStatusEnums . . . . .	128
8.8.2.160 TimerTriggerActivationEnums . . . . .	128
8.8.2.161 TimerTriggerSourceEnums . . . . .	128
8.8.2.162 TransferComponentSelectorEnums . . . . .	129
8.8.2.163 TransferControlModeEnums . . . . .	130
8.8.2.164 TransferOperationModeEnums . . . . .	130
8.8.2.165 TransferQueueModeEnums . . . . .	130
8.8.2.166 TransferSelectorEnums . . . . .	131
8.8.2.167 TransferStatusSelectorEnums . . . . .	131
8.8.2.168 TransferTriggerActivationEnums . . . . .	131

8.8.2.169 TransferTriggerModeEnums . . . . .	132
8.8.2.170 TransferTriggerSelectorEnums . . . . .	132
8.8.2.171 TransferTriggerSourceEnums . . . . .	132
8.8.2.172 TriggerActivationEnums . . . . .	133
8.8.2.173 TriggerModeEnums . . . . .	133
8.8.2.174 TriggerOverlapEnums . . . . .	134
8.8.2.175 TriggerSelectorEnums . . . . .	134
8.8.2.176 TriggerSourceEnums . . . . .	134
8.8.2.177 UserOutputSelectorEnums . . . . .	135
8.8.2.178 UserSetDefaultEnums . . . . .	135
8.8.2.179 UserSetSelectorEnums . . . . .	135
8.8.2.180 WhiteClipSelectorEnums . . . . .	135
8.9 Camera List Class . . . . .	136
8.9.1 Detailed Description . . . . .	136
8.10 CameraPtr Class . . . . .	137
8.10.1 Detailed Description . . . . .	137
8.10.2 Function Documentation . . . . .	137
8.10.2.1 CameraPtr() . . . . .	137
8.10.2.2 CameraPtr(const int) . . . . .	137
8.10.2.3 operator=(const int nMustBeNull) . . . . .	137
8.10.2.4 ~CameraPtr(void) . . . . .	137
8.11 ChunkData Class . . . . .	138
8.11.1 Detailed Description . . . . .	138
8.12 DeviceEvent Class . . . . .	139
8.12.1 Detailed Description . . . . .	139
8.13 Event Class . . . . .	140
8.13.1 Detailed Description . . . . .	140
8.14 Exception Class . . . . .	141
8.14.1 Detailed Description . . . . .	141
8.15 Image Class . . . . .	142

8.15.1 Detailed Description . . . . .	142
8.16 ImageEvent Class . . . . .	143
8.16.1 Detailed Description . . . . .	143
8.17 ImagePtr Class . . . . .	144
8.17.1 Detailed Description . . . . .	144
8.18 ImageStatistics Class . . . . .	145
8.18.1 Detailed Description . . . . .	145
8.19 Interface Class . . . . .	146
8.19.1 Detailed Description . . . . .	146
8.20 InterfaceEvent Class . . . . .	147
8.20.1 Detailed Description . . . . .	147
8.21 InterfaceList Class . . . . .	148
8.21.1 Detailed Description . . . . .	148
8.22 InterfacePtr Class . . . . .	149
8.22.1 Detailed Description . . . . .	149
8.23 LoggingEvent Class . . . . .	150
8.23.1 Detailed Description . . . . .	150
8.24 Logging Event Class . . . . .	151
8.24.1 Detailed Description . . . . .	151
8.25 LoggingEventDataPtr Class . . . . .	152
8.25.1 Detailed Description . . . . .	152
8.26 RemovalEvent Class . . . . .	153
8.26.1 Detailed Description . . . . .	153
8.27 Spinnaker Headers . . . . .	154
8.27.1 Detailed Description . . . . .	155
8.27.2 Variable Documentation . . . . .	155
8.27.2.1 EVENT_TIMEOUT_INFINITE . . . . .	155
8.27.2.2 EVENT_TIMEOUT_NONE . . . . .	155
8.28 Spinnaker.h . . . . .	156
8.29 Spinnaker Definitions . . . . .	157

8.29.1 Detailed Description . . . . .	161
8.29.2 Enumeration Type Documentation . . . . .	161
8.29.2.1 ActionCommandStatus . . . . .	161
8.29.2.2 ColorProcessingAlgorithm . . . . .	161
8.29.2.3 Error . . . . .	162
8.29.2.4 EventType . . . . .	163
8.29.2.5 HeatMapColor . . . . .	163
8.29.2.6 ImageFileFormat . . . . .	163
8.29.2.7 ImageStatus . . . . .	164
8.29.2.8 PayloadTypeInfoIDs . . . . .	164
8.29.2.9 PixelFormatIntType . . . . .	165
8.29.2.10 PixelFormatNamespaceID . . . . .	165
8.29.2.11 PolarizationAlgorithm . . . . .	166
8.29.2.12 PolarizationResolution . . . . .	166
8.29.2.13 SpinnakerLogLevel . . . . .	166
8.29.2.14 StatisticsChannel . . . . .	167
8.30 Spinnaker Platform . . . . .	168
8.30.1 Detailed Description . . . . .	168
8.30.2 Macro Definition Documentation . . . . .	168
8.30.2.1 SPINNAKER_API . . . . .	168
8.30.2.2 SPINNAKER_API_ABSTRACT . . . . .	168
8.30.2.3 SPINNAKER_LOCAL . . . . .	168
8.31 Spinnaker Video Class . . . . .	169
8.31.1 Detailed Description . . . . .	169
8.32 Spinnaker Video Definitions . . . . .	170
8.33 System Class . . . . .	171
8.33.1 Detailed Description . . . . .	171
8.34 SystemPtr Class . . . . .	172
8.34.1 Detailed Description . . . . .	172
8.35 Spinnaker QuickSpin Classes . . . . .	173



8.35.1 Detailed Description . . . . .	173
8.36 TransportLayerDefs Class . . . . .	174
8.36.1 Detailed Description . . . . .	175
8.36.2 Enumeration Type Documentation . . . . .	175
8.36.2.1 DeviceAccessStatusEnum . . . . .	175
8.36.2.2 DeviceCurrentSpeedEnum . . . . .	176
8.36.2.3 DeviceEndiannessMechanismEnum . . . . .	176
8.36.2.4 DeviceTypeEnum . . . . .	176
8.36.2.5 GenICamXMLLocationEnum . . . . .	177
8.36.2.6 GevCCPEnum . . . . .	177
8.36.2.7 GUIXMLLocationEnum . . . . .	177
8.36.2.8 POEStatusEnum . . . . .	177
8.36.2.9 StreamBufferCountModeEnum . . . . .	178
8.36.2.10 StreamBufferHandlingModeEnum . . . . .	178
8.36.2.11 StreamDefaultBufferCountModeEnum . . . . .	178
8.36.2.12 StreamTypeEnum . . . . .	179
8.37 TransportLayerDevice Class . . . . .	180
8.37.1 Detailed Description . . . . .	180
8.38 TransportLayerInterface Class . . . . .	181
8.38.1 Detailed Description . . . . .	181
8.39 TransportLayerStream Class . . . . .	182
8.39.1 Detailed Description . . . . .	182
8.40 Camera Base Interface Class . . . . .	183
8.40.1 Detailed Description . . . . .	183
8.41 IChunkData Class . . . . .	184
8.41.1 Detailed Description . . . . .	184
8.42 IImage Class . . . . .	185
8.42.1 Detailed Description . . . . .	185
8.43 IImageStatistics Class . . . . .	186
8.43.1 Detailed Description . . . . .	186

8.44	Interface Class	187
8.44.1	Detailed Description	187
8.45	InterfaceList Class	188
8.45.1	Detailed Description	188
8.46	ISystem Class	189
8.46.1	Detailed Description	189
8.47	Spinnaker GenApi Classes	190
8.47.1	Detailed Description	196
8.47.2	Typedef Documentation	196
8.47.2.1	CNodeMapRef	196
8.47.2.2	CNodeRef	196
8.47.2.3	CSelectorRef	196
8.47.3	Function Documentation	196
8.47.3.1	_ClearXMLCache()	196
8.47.3.2	_Connect(IPort *pPort, const GenICam::gcstring &PortName)	196
8.47.3.3	_Connect(IPort *pPort)	196
8.47.3.4	_Destroy()	196
8.47.3.5	_GetDeviceName()	196
8.47.3.6	_GetNode(const GenICam::gcstring &key)	196
8.47.3.7	_GetNodes(NodeList_t &Nodes)	196
8.47.3.8	_GetSupportedSchemaVersions(GenICam::gcstring_vector &SchemaVersions)	196
8.47.3.9	_InvalidateNodes()	196
8.47.3.10	_LoadXMLFromFile(const GenICam::gcstring &FileName)	196
8.47.3.11	_LoadXMLFromFileInject(const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)	197
8.47.3.12	_LoadXMLFromString(const GenICam::gcstring &XMLData)	197
8.47.3.13	_LoadXMLFromStringInject(const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)	197
8.47.3.14	_LoadXMLFromZIPData(const void *zipData, size_t zipSize)	197
8.47.3.15	_LoadXMLFromZIPFile(const GenICam::gcstring &ZipFileName)	197
8.47.3.16	_Poll(int64_t ElapsedTime)	197

8.47.3.17	CastToIDestroy(INodeMap *pNodeMap)	197
8.47.3.18	CNodeMapRefT(const GenICam::gcstring &DeviceName=""Device"")	197
8.47.3.19	CNodeMapRefT(INodeMap *pNodeMap, const GenICam::gcstring &DeviceName=""Device"")	197
8.47.3.20	CNodeMapRefT(const CNodeMapRefT &Them)	197
8.47.3.21	EatComments(std::istream &is)	197
8.47.3.22	operator<<(std::ostream &os, const CFeatureBag &FeatureBag)	197
8.47.3.23	operator=(const CNodeMapRefT &Them)	197
8.47.3.24	operator=(INodeMap *pNodeMap)	198
8.47.3.25	operator>>(std::istream &is, CFeatureBag &FeatureBag)	198
8.47.3.26	~CNodeMapRefT()	198
8.48	AutoVector Class	199
8.48.1	Detailed Description	199
8.49	Spinnaker GenApi Interfaces	200
8.49.1	Detailed Description	201
8.49.2	Typedef Documentation	201
8.49.2.1	CallbackHandleType	201
8.49.2.2	NodeList_t	201
8.50	IBase Interface	202
8.50.1	Detailed Description	202
8.50.2	Variable Documentation	202
8.50.2.1	IBase	202
8.51	BooleanNode Class	203
8.51.1	Detailed Description	203
8.51.2	Typedef Documentation	203
8.51.2.1	CBooleanRef	203
8.52	CategoryNode Class	204
8.52.1	Detailed Description	204
8.52.2	Typedef Documentation	204
8.52.2.1	CCategoryRef	204
8.53	ChunkAdapter Class	205

8.53.1 Detailed Description . . . . .	205
8.54 ChunkAdapterDcam Class . . . . .	206
8.54.1 Detailed Description . . . . .	206
8.55 ChunkAdapterGeneric Class . . . . .	207
8.55.1 Detailed Description . . . . .	207
8.56 ChunkAdapterGEV Class . . . . .	208
8.56.1 Detailed Description . . . . .	208
8.57 ChunkPort Class . . . . .	209
8.57.1 Detailed Description . . . . .	209
8.58 CommandNode Class . . . . .	210
8.58.1 Detailed Description . . . . .	210
8.58.2 Typedef Documentation . . . . .	210
8.58.2.1 CCommandRef . . . . .	210
8.59 Container Class . . . . .	211
8.60 Counter Class . . . . .	212
8.60.1 Detailed Description . . . . .	212
8.61 EnumClasses Class . . . . .	213
8.61.1 Detailed Description . . . . .	214
8.62 EnumEntryNode Class . . . . .	215
8.62.1 Detailed Description . . . . .	215
8.62.2 Typedef Documentation . . . . .	215
8.62.2.1 CEnumEntryRef . . . . .	215
8.63 EnumNode Class . . . . .	216
8.63.1 Detailed Description . . . . .	216
8.63.2 Typedef Documentation . . . . .	216
8.63.2.1 CEnumerationRef . . . . .	216
8.64 EnumNodeT Class . . . . .	217
8.64.1 Detailed Description . . . . .	217
8.65 EventAdapter Class . . . . .	218
8.65.1 Detailed Description . . . . .	218

8.66	EventAdapter1394 Class	219
8.66.1	Detailed Description	219
8.67	EventAdapterGeneric Class	220
8.67.1	Detailed Description	220
8.68	EventAdapterGEV Class	221
8.68.1	Detailed Description	221
8.69	EventAdapterU3V Class	222
8.69.1	Detailed Description	222
8.70	EventPort Class	223
8.70.1	Detailed Description	223
8.71	Filestream Class	224
8.71.1	Detailed Description	224
8.72	FloatNode Class	225
8.72.1	Detailed Description	225
8.72.2	Typedef Documentation	225
8.72.2.1	CFloatRef	225
8.73	FloatRegNode Class	226
8.73.1	Detailed Description	226
8.74	GCString Class	227
8.74.1	Detailed Description	227
8.75	GCSynch Class	228
8.75.1	Detailed Description	228
8.76	GCTypes Class	229
8.76.1	Detailed Description	229
8.76.2	Typedef Documentation	229
8.76.2.1	float32_t	229
8.76.2.2	float64_t	229
8.77	Spinnaker GenApi Utilities	230
8.77.1	Detailed Description	230
8.78	GCUtilities Utility	231

8.78.1 Detailed Description	232
8.78.2 Function Documentation	232
8.78.2.1 DoesEnvironmentVariableExist(const Spinnaker::GenICam::gcstring &VariableName)	232
8.78.2.2 GetFiles(const gcstring &FileTemplate, gcstring_vector &FileNames, const bool DirectoriesOnly=false)	232
8.78.2.3 GetGenICamCacheFolder(void)	232
8.78.2.4 GetGenICamCLProtocolFolder(void)	232
8.78.2.5 GetGenICamLogConfig(void)	233
8.78.2.6 GetModulePathFromFunction(void *pFunction)	233
8.78.2.7 GetValueOfEnvironmentVariable(const gcstring &VariableName)	233
8.78.2.8 GetValueOfEnvironmentVariable(const gcstring &VariableName, gcstring &VariableContent)	233
8.78.2.9 INTEGRAL_CAST(int64_t ll)	233
8.78.2.10 INTEGRAL_CAST2(Ts s)	233
8.78.2.11 ReplaceEnvironmentVariables(gcstring &Buffer, bool ReplaceBlankBy20=false)	234
8.78.2.12 SetGenICamCacheFolder(const gcstring &path)	234
8.78.2.13 SetGenICamCLProtocolFolder(const gcstring &path)	234
8.78.2.14 SetGenICamLogConfig(const gcstring &path)	234
8.78.2.15 Tokenize(const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters="" "")	234
8.78.2.16 UrlDecode(const gcstring &Input)	234
8.78.2.17 UrlEncode(const gcstring &Input)	234
8.79 IBoolean Interface	235
8.79.1 Detailed Description	235
8.79.2 Function Documentation	235
8.79.2.1 GetValue(bool Verify=false, bool IgnoreCache=false) const =0	235
8.79.2.2 operator>() const	236
8.79.2.3 operator=(bool Value)	236
8.79.3 Variable Documentation	236
8.79.3.1 IBoolean	236
8.79.3.2 Verify	236

8.80	ICategory Interfaces	237
8.80.1	Detailed Description	237
8.80.2	Variable Documentation	237
8.80.2.1	ICategory	237
8.81	ICChunkPort Interface	238
8.81.1	Detailed Description	238
8.81.2	Macro Definition Documentation	238
8.81.2.1	CHUNK_BASE_ADDRESS_REGISTER	238
8.81.2.2	CHUNK_BASE_ADDRESS_REGISTER_LEN	238
8.81.2.3	CHUNK_LENGTH_REGISTER	239
8.81.2.4	CHUNK_LENGTH_REGISTER_LEN	239
8.81.3	Function Documentation	239
8.81.3.1	CacheChunkData() const =0	239
8.81.4	Variable Documentation	239
8.81.4.1	ICChunkPort	239
8.82	ICommand Interface	240
8.82.1	Detailed Description	240
8.82.2	Function Documentation	240
8.82.2.1	IsDone(bool Verify=true)=0	240
8.82.3	Variable Documentation	240
8.82.3.1	ICommand	240
8.83	IDestroy Interface	241
8.83.1	Detailed Description	241
8.83.2	Variable Documentation	241
8.83.2.1	IDestroy	241
8.84	IDeviceInfo Interface	242
8.84.1	Detailed Description	242
8.84.2	Function Documentation	242
8.84.2.1	GetDeviceVersion(GenICam::Version_t &Version)=0	242
8.84.2.2	GetGenApiVersion(GenICam::Version_t &Version, uint16_t &Build)=0	243

8.84.2.3	GetProductGuid()=0	243
8.84.2.4	GetSchemaVersion(GenICam::Version_t &Version)=0	243
8.84.2.5	GetStandardNameSpace()=0	243
8.84.2.6	GetToolTip()=0	243
8.84.2.7	GetVendorName()=0	243
8.84.2.8	GetVersionGuid()=0	243
8.84.3	Variable Documentation	243
8.84.3.1	IDeviceInfo	243
8.85	IEnumEntry Interface	244
8.85.1	Detailed Description	244
8.85.2	Function Documentation	244
8.85.2.1	GetNumericValue()=0	244
8.85.2.2	GetSymbolic() const =0	244
8.85.2.3	IsSelfClearing()=0	244
8.85.3	Variable Documentation	244
8.85.3.1	IEnumEntry	244
8.86	IEnumeration Interface	245
8.86.1	Detailed Description	245
8.86.2	Function Documentation	245
8.86.2.1	GetCurrentEntry(bool Verify=false, bool IgnoreCache=false)=0	245
8.86.2.2	GetEntries(NodeList_t &Entries)=0	245
8.86.2.3	GetEntry(const int64_t IntValue)=0	246
8.86.2.4	GetEntryByName(const GenICam::gcstring &Symbolic)=0	246
8.86.2.5	GetIntValue(bool Verify=false, bool IgnoreCache=false)=0	246
8.86.2.6	operator*()=0	246
8.86.2.7	SetIntValue(int64_t Value, bool Verify=true)=0	246
8.86.3	Variable Documentation	246
8.86.3.1	IEnumeration	246
8.87	IEnumerationT Interface	247
8.87.1	Detailed Description	247



8.87.2	Function Documentation	247
8.87.2.1	GetEntry(const EnumT Value)=0	247
8.87.2.2	operator=(EnumT Value)=0	247
8.87.2.3	operator=(const GenICam::gcstring &ValueStr)=0	248
8.87.3	Variable Documentation	248
8.87.3.1	IEnumerationT	248
8.87.3.2	IEnumReference	248
8.88	IFloat Interface	249
8.88.1	Detailed Description	250
8.88.2	Function Documentation	250
8.88.2.1	GetDisplayNotation() const =0	250
8.88.2.2	GetDisplayPrecision() const =0	250
8.88.2.3	GetInc()=0	250
8.88.2.4	GetIncMode()=0	250
8.88.2.5	GetListOfValidValues(bool bounded=true)=0	250
8.88.2.6	GetMax()=0	250
8.88.2.7	GetMin()=0	250
8.88.2.8	GetRepresentation()=0	250
8.88.2.9	GetUnit() const =0	250
8.88.2.10	HasInc()=0	251
8.88.2.11	ImposeMax(double Value)=0	251
8.88.2.12	ImposeMin(double Value)=0	251
8.88.2.13	operator=(double Value)=0	251
8.88.3	Variable Documentation	251
8.88.3.1	IFloat	251
8.89	Integer Interface	252
8.89.1	Detailed Description	252
8.89.2	Function Documentation	252
8.89.2.1	ImposeMax(int64_t Value)=0	252
8.89.2.2	ImposeMin(int64_t Value)=0	252

8.89.2.3	operator=(int64_t Value)=0	252
8.89.3	Variable Documentation	252
8.89.3.1	Integer	252
8.90	INode Interface	253
8.90.1	Detailed Description	255
8.90.2	Function Documentation	255
8.90.2.1	Combine(EAccessMode Peter, EAccessMode Paul)	255
8.90.2.2	Combine(EVisibility Peter, EVisibility Paul)	255
8.90.2.3	Combine(ECachingMode Peter, ECachingMode Paul)	255
8.90.2.4	DeregisterCallback(CallbackHandleType hCallback)=0	255
8.90.2.5	GetAlias() const =0	255
8.90.2.6	GetCachingMode() const =0	255
8.90.2.7	GetCastAlias() const =0	255
8.90.2.8	GetChildren(GenApi::NodeList_t &Children, ELinkType LinkType=ctReading↵ Children) const =0	255
8.90.2.9	GetDescription() const =0	256
8.90.2.10	GetDisplayName() const =0	256
8.90.2.11	GetDocuURL() const =0	256
8.90.2.12	GetEventID() const =0	256
8.90.2.13	GetNameSpace() const =0	256
8.90.2.14	GetNodeMap() const =0	256
8.90.2.15	GetParents(GenApi::NodeList_t &Parents) const =0	256
8.90.2.16	GetPollingTime() const =0	256
8.90.2.17	GetPrincipalInterfaceType() const =0	257
8.90.2.18	GetProperty(const GenICam::gcstring &PropertyName, GenICam::gcstring &↵ ValueStr, GenICam::gcstring &AttributeStr)=0	257
8.90.2.19	GetPropertyNames(GenICam::gcstring_vector &PropertyNames) const =0	257
8.90.2.20	GetVisibility() const =0	257
8.90.2.21	ImposeAccessMode(EAccessMode ImposedAccessMode)=0	257
8.90.2.22	ImposeVisibility(EVisibility ImposedVisibility)=0	257
8.90.2.23	InvalidateNode()=0	257

8.90.2.24	IsAccessModeCacheable() const =0	257
8.90.2.25	IsAvailable(EAccessMode AccessMode)	257
8.90.2.26	IsAvailable(const IBase *p)	258
8.90.2.27	IsAvailable(const IBase &r)	258
8.90.2.28	IsCacheable() const =0	258
8.90.2.29	IsCacheable(ECachingMode CachingMode)	258
8.90.2.30	IsDeprecated() const =0	258
8.90.2.31	IsFeature() const =0	258
8.90.2.32	IsImplemented(EAccessMode AccessMode)	258
8.90.2.33	IsImplemented(const IBase *p)	258
8.90.2.34	IsImplemented(const IBase &r)	258
8.90.2.35	IsReadable(EAccessMode AccessMode)	258
8.90.2.36	IsReadable(const IBase *p)	259
8.90.2.37	IsReadable(const IBase &r)	259
8.90.2.38	IsStreamable() const =0	259
8.90.2.39	IsVisible(EVisibility Visibility, EVisibility MaxVisiblity)	259
8.90.2.40	IsWritable(EAccessMode AccessMode)	259
8.90.2.41	IsWritable(const IBase *p)	259
8.90.2.42	IsWritable(const IBase &r)	259
8.90.2.43	operator!=(int nullPtr) const =0	259
8.90.2.44	operator==(int nullPtr) const =0	259
8.90.2.45	RegisterCallback(CNodeCallback *pCallback)=0	259
8.90.3	Variable Documentation	259
8.90.3.1	INode	259
8.90.3.2	IReference	259
8.91	INodeMap Interface	260
8.91.1	Detailed Description	260
8.91.2	Function Documentation	260
8.91.2.1	Connect(IPort *pPort, const GenICam::gcstring &PortName) const =0	260
8.91.2.2	Connect(IPort *pPort) const =0	261

8.91.2.3	GetDeviceName() const =0	261
8.91.2.4	GetLock() const =0	261
8.91.2.5	GetNode(const GenICam::gcstring &Name) const =0	261
8.91.2.6	GetNumNodes() const =0	261
8.91.2.7	InvalidateNodes() const =0	261
8.91.2.8	Poll(int64_t ElapsedTime)=0	261
8.91.3	Variable Documentation	261
8.91.3.1	INodeMap	261
8.92	INodeMapDyn Interface	262
8.92.1	Detailed Description	263
8.92.2	Function Documentation	263
8.92.2.1	ExtractIndependentSubtree(const GenICam::gcstring &XMLData, const GenICam::gcstring &InjectXMLData, const GenICam::gcstring &SubTreeRootNodeName, GenICam::gcstring &ExtractedSubtree)=0	263
8.92.2.2	GetSupportedSchemaVersions(GenICam::gcstring_vector &SchemaVersions)=0	263
8.92.2.3	LoadXMLFromFile(const GenICam::gcstring &FileName)=0	263
8.92.2.4	LoadXMLFromFileInject(const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)=0	263
8.92.2.5	LoadXMLFromString(const GenICam::gcstring &XMLData)=0	263
8.92.2.6	LoadXMLFromStringInject(const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)=0	264
8.92.2.7	LoadXMLFromZIPData(const void *zipData, size_t zipSize)=0	264
8.92.2.8	LoadXMLFromZIPFile(const GenICam::gcstring &ZipFileName)=0	264
8.92.2.9	MergeXMLFiles(const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectedFileName, const GenICam::gcstring &OutputFileName)=0	264
8.92.2.10	PreprocessXMLFromFile(const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xvDefault)=0	264
8.92.2.11	PreprocessXMLFromZIPFile(const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32_t XMLValidation=xvDefault)=0	265
8.92.3	Variable Documentation	265
8.92.3.1	INodeMapDyn	265
8.93	IntegerNode Class	266

8.93.1 Detailed Description . . . . .	266
8.93.2 Typedef Documentation . . . . .	266
8.93.2.1 CIntegerRef . . . . .	266
8.94 IntRegNode Class . . . . .	267
8.94.1 Detailed Description . . . . .	267
8.95 IPort Interface . . . . .	268
8.95.1 Detailed Description . . . . .	268
8.95.2 Function Documentation . . . . .	268
8.95.2.1 Write(const void *pBuffer, int64_t Address, int64_t Length)=0 . . . . .	268
8.95.3 Variable Documentation . . . . .	268
8.95.3.1 Address . . . . .	268
8.95.3.2 IPort . . . . .	268
8.95.3.3 Length . . . . .	268
8.96 IPortConstruct Interface . . . . .	269
8.96.1 Detailed Description . . . . .	269
8.96.2 Function Documentation . . . . .	269
8.96.2.1 GetSwapEndianess()=0 . . . . .	269
8.96.3 Variable Documentation . . . . .	269
8.96.3.1 IPortConstruct . . . . .	269
8.97 IPortRecorder Interface . . . . .	270
8.97.1 Detailed Description . . . . .	270
8.97.2 Function Documentation . . . . .	270
8.97.2.1 GetCookie()=0 . . . . .	270
8.97.2.2 Replay(IPort *pPort)=0 . . . . .	270
8.97.2.3 SetCookie(const int64_t Value)=0 . . . . .	271
8.97.2.4 StopRecording()=0 . . . . .	271
8.97.3 Variable Documentation . . . . .	271
8.97.3.1 Invalidate . . . . .	271
8.97.3.2 IPortRecorder . . . . .	271
8.97.3.3 IPortReplay . . . . .	271

8.97.3.4	IPortWriteList	271
8.98	IRegister Interfaces	272
8.98.1	Detailed Description	272
8.98.2	Function Documentation	272
8.98.2.1	Get(uint8_t *pBuffer, int64_t Length, bool Verify=false, bool IgnoreCache=false)=0	272
8.98.2.2	GetAddress()=0	273
8.98.2.3	GetLength()=0	273
8.98.3	Variable Documentation	273
8.98.3.1	IRegister	273
8.99	ISelector Interface	274
8.99.1	Detailed Description	274
8.99.2	Function Documentation	274
8.99.2.1	GetSelectedFeatures(FeatureList_t &) const =0	274
8.99.2.2	GetSelectingFeatures(FeatureList_t &) const =0	274
8.99.3	Variable Documentation	274
8.99.3.1	ISelector	274
8.100	ISelectorDigit Interface	275
8.100.1	Detailed Description	275
8.100.2	Function Documentation	275
8.100.2.1	GetSelectorList(FeatureList_t &SelectorList, bool Incremental=false)=0	275
8.100.2.2	Restore()=0	276
8.100.2.3	SetNext(bool Tick=true)=0	276
8.100.2.4	ToString()=0	276
8.100.3	Variable Documentation	276
8.100.3.1	ISelectorDigit	276
8.101	IString Class	277
8.101.1	Detailed Description	277
8.101.2	Function Documentation	277
8.101.2.1	GetMaxLength()=0	277
8.101.3	Variable Documentation	277

8.101.3.1 IString . . . . .	277
8.102IValue Class . . . . .	278
8.102.1 Detailed Description . . . . .	278
8.102.2 Function Documentation . . . . .	278
8.102.2.1 FromString(const GenICam::gcstring &ValueStr, bool Verify=true)=0 . . . . .	278
8.102.2.2 IsValueCacheValid() const =0 . . . . .	278
8.102.2.3 ToString(bool Verify=false, bool IgnoreCache=false)=0 . . . . .	279
8.102.3 Variable Documentation . . . . .	279
8.102.3.1 IValue . . . . .	279
8.103Node Class . . . . .	280
8.103.1 Detailed Description . . . . .	280
8.104NodeCallback Class . . . . .	281
8.104.1 Detailed Description . . . . .	282
8.104.2 Enumeration Type Documentation . . . . .	282
8.104.2.1 ECallbackType . . . . .	282
8.104.3 Function Documentation . . . . .	282
8.104.3.1 Deregister(GenApi::CallbackHandleType pCallbackInfo) . . . . .	282
8.104.3.2 make_NodeCallback(INode *pNode, Function function, ECallbackType CallbackType) . . . . .	282
8.104.3.3 make_NodeCallback(INode *pNode, Client &client, Member member, E↔ CallbackType CallbackType) . . . . .	282
8.104.3.4 Register(INode *pNode, Function f, ECallbackType CallbackType=cbPostInside↔ Lock) . . . . .	282
8.104.3.5 Register(INode *pNode, Client &c, Member m, ECallbackType Callback↔ Type=cbPostInsideLock) . . . . .	282
8.105NodeMap Class . . . . .	283
8.105.1 Detailed Description . . . . .	283
8.106NodeMapFactory Class . . . . .	284
8.106.1 Detailed Description . . . . .	284
8.106.2 Enumeration Type Documentation . . . . .	284
8.106.2.1 ECacheUsage_t . . . . .	284
8.106.2.2 EContentType_t . . . . .	285

8.107NodeMapRef Class . . . . .	286
8.107.1 Detailed Description . . . . .	286
8.108Persistence Class . . . . .	287
8.108.1 Detailed Description . . . . .	287
8.109Pointer Class . . . . .	288
8.109.1 Detailed Description . . . . .	289
8.109.2 Typedef Documentation . . . . .	289
8.109.2.1 CBasePtr . . . . .	289
8.109.2.2 CBooleanPtr . . . . .	289
8.109.2.3 CCategoryPtr . . . . .	290
8.109.2.4 CChunkPortPtr . . . . .	290
8.109.2.5 CCommandPtr . . . . .	290
8.109.2.6 CDeviceInfoPtr . . . . .	290
8.109.2.7 CEnumEntryPtr . . . . .	290
8.109.2.8 CEnumerationPtr . . . . .	290
8.109.2.9 CIntegerPtr . . . . .	290
8.109.2.10CNodeMapDynPtr . . . . .	290
8.109.2.11CNodeMapPtr . . . . .	290
8.109.2.12CNodePtr . . . . .	290
8.109.2.13CPortConstructPtr . . . . .	291
8.109.2.14CPortPtr . . . . .	291
8.109.2.15CPortRecorderPtr . . . . .	291
8.109.2.16CPortReplayPtr . . . . .	291
8.109.2.17CPortWriteListPtr . . . . .	291
8.109.2.18CRegisterPtr . . . . .	291
8.109.2.19CSelectorPtr . . . . .	291
8.109.2.20CStringPtr . . . . .	291
8.109.2.21CValuePtr . . . . .	291
8.109.3 Function Documentation . . . . .	291
8.109.3.1 GetInterfaceName(IBase *pBase) . . . . .	291



8.109.3.2	IsAvailable(const Spinnaker::GenApi::CPointer< T, B > &ptr)	292
8.109.3.3	IsImplemented(const Spinnaker::GenApi::CPointer< T, B > &ptr)	292
8.109.3.4	IsReadable(const Spinnaker::GenApi::CPointer< T, B > &ptr)	292
8.109.3.5	IsWritable(const Spinnaker::GenApi::CPointer< T, B > &ptr)	292
8.110	PortImpl Class	293
8.110.1	Detailed Description	293
8.111	PortNode Class	294
8.111.1	Detailed Description	294
8.111.2	Typedef Documentation	294
8.111.2.1	CPortRef	294
8.112	PortRecorder Class	295
8.112.1	Detailed Description	295
8.112.2	Typedef Documentation	295
8.112.2.1	CPortRecorderRef	295
8.113	PortReplay Class	296
8.113.1	Detailed Description	296
8.114	PortWriteList Class	297
8.114.1	Detailed Description	297
8.115	Reference Interfaces	298
8.115.1	Detailed Description	298
8.115.2	Function Documentation	298
8.115.2.1	SetNumEnums(int NumEnums)=0	298
8.116	RegisterNode Class	299
8.116.1	Detailed Description	299
8.116.2	Typedef Documentation	299
8.116.2.1	CRegisterRef	299
8.117	RegisterPortImpl Class	300
8.117.1	Detailed Description	300
8.118	SelectorSet Class	301
8.118.1	Detailed Description	301

8.119SpinTestCamera Class . . . . .	302
8.119.1 Detailed Description . . . . .	302
8.120StringNode Class . . . . .	303
8.120.1 Detailed Description . . . . .	303
8.120.2 Typedef Documentation . . . . .	303
8.120.2.1 CStringRef . . . . .	303
8.121StringRegNode Class . . . . .	304
8.121.1 Detailed Description . . . . .	304
8.122StructPort Class . . . . .	305
8.122.1 Detailed Description . . . . .	305
8.123Synch Class . . . . .	306
8.123.1 Detailed Description . . . . .	306
8.124Spinnaker GenApi Enums . . . . .	307
8.124.1 Detailed Description . . . . .	307
8.125Types Enums . . . . .	308
8.125.1 Detailed Description . . . . .	310
8.125.2 Macro Definition Documentation . . . . .	310
8.125.2.1 _UndefinedRepresentation . . . . .	310
8.125.3 Typedef Documentation . . . . .	310
8.125.3.1 StringList_t . . . . .	310
8.125.4 Enumeration Type Documentation . . . . .	311
8.125.4.1 EAccessMode . . . . .	311
8.125.4.2 ECachingMode . . . . .	311
8.125.4.3 EDisplayNotation . . . . .	311
8.125.4.4 EEndianess . . . . .	311
8.125.4.5 EGenApiSchemaVersion . . . . .	312
8.125.4.6 EIncMode . . . . .	312
8.125.4.7 EInputDirection . . . . .	312
8.125.4.8 EInterfaceType . . . . .	312
8.125.4.9 ELinkType . . . . .	313

8.125.4.10	ENamespace	313
8.125.4.11	ERepresentation	313
8.125.4.12	ESign	313
8.125.4.13	ESlope	314
8.125.4.14	EStandardNamespace	314
8.125.4.15	EVisibility	314
8.125.4.16	EXMLValidation	314
8.125.4.17	EYesNo	315
8.126	ValueNode Class	316
8.126.1	Detailed Description	316
8.126.2	Typedef Documentation	316
8.126.2.1	CValueRef	316
8.127	ChunkAdapterU3V Class	317
8.127.1	Detailed Description	317
<b>9</b>	<b>Namespace Documentation</b>	<b>319</b>
9.1	Spinnaker Namespace Reference	319
9.2	Spinnaker::GenApi Namespace Reference	356
9.2.1	Typedef Documentation	371
9.2.1.1	IDevFileStream	371
9.2.1.2	ODevFileStream	371
9.2.2	Enumeration Type Documentation	371
9.2.2.1	GVCP_MESSAGE_TAGS	371
9.2.3	Function Documentation	371
9.2.3.1	PersistFeature(IValue &item)=0	371
9.2.3.2	SET_GUID(SPIN_GUID &name, uint32_t l, uint16_t w1, uint16_t w2, uint8_t b1, uint8_t b2, uint8_t b3, uint8_t b4, uint8_t b5, uint8_t b6, uint8_t b7, uint8_t b8)	372
9.2.4	Variable Documentation	372
9.2.4.1	COMMAND_MAGIC	372
9.2.4.2	GENCP_COMMAND_HEADER_SIZE	372
9.2.4.3	GENCP_EVENT_BASIC_SIZE	372
9.2.4.4	GENCP_EVENT_CMD_ID	372
9.2.4.5	IPersistScript	372
9.2.4.6	U3V_EVENT_PREFIX	372
9.3	Spinnaker::GenICam Namespace Reference	372
9.3.1	Function Documentation	374
9.3.1.1	getline(std::istream &is, Spinnaker::GenICam::gcstring &str)	374
9.3.1.2	getline(std::istream &is, Spinnaker::GenICam::gcstring &str, char delim)	374
9.3.1.3	ThrowBadAlloc()	374
9.4	Spinnaker::Video Namespace Reference	374

<b>10 Class Documentation</b>	<b>375</b>
10.1 ActionCommandResult Struct Reference	375
10.1.1 Detailed Description	375
10.1.2 Member Data Documentation	375
10.1.2.1 DeviceAddress	375
10.1.2.2 Status	375
10.2 ArrivalEvent Class Reference	376
10.2.1 Detailed Description	377
10.2.2 Constructor & Destructor Documentation	377
10.2.2.1 ArrivalEvent()	377
10.2.2.2 ~ArrivalEvent()	377
10.2.3 Member Function Documentation	377
10.2.3.1 OnDeviceArrival(uint64_t serialNumber)=0	377
10.2.3.2 operator=(const ArrivalEvent &)	377
10.3 AttachStatistics_t Struct Reference	377
10.3.1 Detailed Description	378
10.3.2 Member Data Documentation	378
10.3.2.1 NumAttachedChunks	378
10.3.2.2 NumChunkPorts	378
10.3.2.3 NumChunks	378
10.4 AutoLock Class Reference	378
10.4.1 Constructor & Destructor Documentation	378
10.4.1.1 AutoLock(CLock &lock)	378
10.4.1.2 ~AutoLock()	378
10.5 AutoLock Class Reference	379
10.5.1 Constructor & Destructor Documentation	379
10.5.1.1 AutoLock(CLock &lock)	379
10.5.1.2 ~AutoLock()	379
10.6 AVIOption Struct Reference	379
10.6.1 Detailed Description	379

10.6.2	Constructor & Destructor Documentation . . . . .	379
10.6.2.1	AVIOption() . . . . .	379
10.6.3	Member Data Documentation . . . . .	379
10.6.3.1	frameRate . . . . .	379
10.6.3.2	reserved . . . . .	380
10.7	BasePtr< T, B > Class Template Reference . . . . .	380
10.7.1	Detailed Description . . . . .	380
10.7.2	Constructor & Destructor Documentation . . . . .	381
10.7.2.1	BasePtr(void) . . . . .	381
10.7.2.2	~BasePtr(void) . . . . .	381
10.7.2.3	BasePtr(const BasePtr &other) . . . . .	381
10.7.3	Member Function Documentation . . . . .	381
10.7.3.1	get() const . . . . .	381
10.7.3.2	IsValid() const . . . . .	381
10.7.3.3	operator bool(void) const . . . . .	381
10.7.3.4	operator T *(void) const . . . . .	381
10.7.3.5	operator->(void) const . . . . .	381
10.7.3.6	operator=(const BasePtr &rhs) . . . . .	381
10.7.3.7	operator=(const int nMustBeNull) . . . . .	381
10.7.3.8	operator==(const BasePtr &rT) const . . . . .	382
10.7.3.9	operator==(int nMustBeNull) const . . . . .	382
10.7.4	Member Data Documentation . . . . .	382
10.7.4.1	m_pT . . . . .	382
10.8	BMPOption Struct Reference . . . . .	382
10.8.1	Detailed Description . . . . .	382
10.8.2	Constructor & Destructor Documentation . . . . .	383
10.8.2.1	BMPOption() . . . . .	383
10.8.3	Member Data Documentation . . . . .	383
10.8.3.1	indexedColor_8bit . . . . .	383
10.8.3.2	reserved . . . . .	383

10.9 BooleanNode Class Reference . . . . .	383
10.9.1 Detailed Description . . . . .	384
10.9.2 Constructor & Destructor Documentation . . . . .	385
10.9.2.1 BooleanNode() . . . . .	385
10.9.2.2 BooleanNode(std::shared_ptr< Node::NodeImpl > pBoolean) . . . . .	385
10.9.2.3 ~BooleanNode() . . . . .	385
10.9.3 Member Function Documentation . . . . .	385
10.9.3.1 GetValue(bool Verify=false, bool IgnoreCache=false) const . . . . .	385
10.9.3.2 operator=(bool Value) . . . . .	385
10.9.3.3 SetReference(INode *pBase) . . . . .	385
10.9.3.4 SetValue(bool Value, bool Verify=true) . . . . .	385
10.10 Camera Class Reference . . . . .	386
10.10.1 Detailed Description . . . . .	416
10.10.2 Constructor & Destructor Documentation . . . . .	416
10.10.2.1 ~Camera() . . . . .	416
10.10.2.2 Camera() . . . . .	416
10.10.3 Member Function Documentation . . . . .	416
10.10.3.1 Init() . . . . .	416
10.10.4 Member Data Documentation . . . . .	416
10.10.4.1 AasRoiEnable . . . . .	416
10.10.4.2 AasRoiHeight . . . . .	416
10.10.4.3 AasRoiOffsetX . . . . .	416
10.10.4.4 AasRoiOffsetY . . . . .	417
10.10.4.5 AasRoiWidth . . . . .	417
10.10.4.6 AcquisitionAbort . . . . .	417
10.10.4.7 AcquisitionArm . . . . .	417
10.10.4.8 AcquisitionBurstFrameCount . . . . .	417
10.10.4.9 AcquisitionFrameCount . . . . .	418
10.10.4.10 AcquisitionFrameRate . . . . .	418
10.10.4.11 AcquisitionFrameRateEnable . . . . .	418

10.10.4.12AcquisitionLineRate . . . . .	418
10.10.4.13AcquisitionMode . . . . .	418
10.10.4.14AcquisitionResultingFrameRate . . . . .	418
10.10.4.15AcquisitionStart . . . . .	418
10.10.4.16AcquisitionStatus . . . . .	419
10.10.4.17AcquisitionStatusSelector . . . . .	419
10.10.4.18AcquisitionStop . . . . .	419
10.10.4.19ActionDeviceKey . . . . .	419
10.10.4.20ActionGroupKey . . . . .	419
10.10.4.21ActionGroupMask . . . . .	419
10.10.4.22ActionQueueSize . . . . .	419
10.10.4.23ActionSelector . . . . .	420
10.10.4.24ActionUnconditionalMode . . . . .	420
10.10.4.25AdaptiveCompressionEnable . . . . .	420
10.10.4.26AdcBitDepth . . . . .	420
10.10.4.27aPAUSEMACCtrlFramesReceived . . . . .	420
10.10.4.28aPAUSEMACCtrlFramesTransmitted . . . . .	420
10.10.4.29AutoAlgorithmSelector . . . . .	420
10.10.4.30AutoExposureControlLoopDamping . . . . .	421
10.10.4.31AutoExposureControlPriority . . . . .	421
10.10.4.32AutoExposureEVCompensation . . . . .	421
10.10.4.33AutoExposureExposureTimeLowerLimit . . . . .	421
10.10.4.34AutoExposureExposureTimeUpperLimit . . . . .	421
10.10.4.35AutoExposureGainLowerLimit . . . . .	422
10.10.4.36AutoExposureGainUpperLimit . . . . .	422
10.10.4.37AutoExposureGreyValueLowerLimit . . . . .	422
10.10.4.38AutoExposureGreyValueUpperLimit . . . . .	422
10.10.4.39AutoExposureLightingMode . . . . .	422
10.10.4.40AutoExposureMeteringMode . . . . .	423
10.10.4.41AutoExposureTargetGreyValue . . . . .	423

10.10.4.42AutoExposureTargetGreyValueAuto . . . . .	423
10.10.4.43BalanceRatio . . . . .	423
10.10.4.44BalanceRatioSelector . . . . .	424
10.10.4.45BalanceWhiteAuto . . . . .	424
10.10.4.46BalanceWhiteAutoDamping . . . . .	424
10.10.4.47BalanceWhiteAutoLowerLimit . . . . .	424
10.10.4.48BalanceWhiteAutoProfile . . . . .	424
10.10.4.49BalanceWhiteAutoUpperLimit . . . . .	425
10.10.4.50BinningHorizontal . . . . .	425
10.10.4.51BinningHorizontalMode . . . . .	425
10.10.4.52BinningSelector . . . . .	425
10.10.4.53BinningVertical . . . . .	425
10.10.4.54BinningVerticalMode . . . . .	425
10.10.4.55BlackLevel . . . . .	426
10.10.4.56BlackLevelAuto . . . . .	426
10.10.4.57BlackLevelAutoBalance . . . . .	426
10.10.4.58BlackLevelClampingEnable . . . . .	426
10.10.4.59BlackLevelRaw . . . . .	426
10.10.4.60BlackLevelSelector . . . . .	426
10.10.4.61ChunkBlackLevel . . . . .	427
10.10.4.62ChunkBlackLevelSelector . . . . .	427
10.10.4.63ChunkCounterSelector . . . . .	427
10.10.4.64ChunkCounterValue . . . . .	427
10.10.4.65ChunkCRC . . . . .	427
10.10.4.66ChunkEnable . . . . .	427
10.10.4.67ChunkEncoderSelector . . . . .	427
10.10.4.68ChunkEncoderStatus . . . . .	427
10.10.4.69ChunkEncoderValue . . . . .	428
10.10.4.70ChunkExposureEndLineStatusAll . . . . .	428
10.10.4.71ChunkExposureTime . . . . .	428



10.10.4.72	ChunkExposureTimeSelector	428
10.10.4.73	ChunkFrameID	428
10.10.4.74	ChunkGain	428
10.10.4.75	ChunkGainSelector	428
10.10.4.76	ChunkHeight	428
10.10.4.77	ChunkImage	429
10.10.4.78	ChunkImageComponent	429
10.10.4.79	ChunkInferenceConfidence	429
10.10.4.80	ChunkInferenceResult	429
10.10.4.81	ChunkLinePitch	429
10.10.4.82	ChunkLineStatusAll	429
10.10.4.83	ChunkModeActive	429
10.10.4.84	ChunkOffsetX	429
10.10.4.85	ChunkOffsetY	430
10.10.4.86	ChunkPartSelector	430
10.10.4.87	ChunkPixelDynamicRangeMax	430
10.10.4.88	ChunkPixelDynamicRangeMin	430
10.10.4.89	ChunkPixelFormat	430
10.10.4.90	ChunkRegionID	430
10.10.4.91	ChunkScan3dAxisMax	430
10.10.4.92	ChunkScan3dAxisMin	430
10.10.4.93	ChunkScan3dCoordinateOffset	431
10.10.4.94	ChunkScan3dCoordinateReferenceSelector	431
10.10.4.95	ChunkScan3dCoordinateReferenceValue	431
10.10.4.96	ChunkScan3dCoordinateScale	431
10.10.4.97	ChunkScan3dCoordinateSelector	431
10.10.4.98	ChunkScan3dCoordinateSystem	431
10.10.4.99	ChunkScan3dCoordinateSystemReference	431
10.10.4.100	ChunkScan3dCoordinateTransformSelector	432
10.10.4.100	ChunkScan3dDistanceUnit	432

10.10.4.102	hunkScan3dInvalidDataFlag	432
10.10.4.103	hunkScan3dInvalidDataValue	432
10.10.4.104	hunkScan3dOutputMode	432
10.10.4.105	hunkScan3dTransformValue	432
10.10.4.106	hunkScanLineSelector	432
10.10.4.107	hunkSelector	433
10.10.4.108	hunkSequencerSetActive	433
10.10.4.109	hunkSerialData	433
10.10.4.110	hunkSerialDataLength	433
10.10.4.111	hunkSerialReceiveOverflow	433
10.10.4.112	hunkSourceID	433
10.10.4.113	hunkStreamChannelID	433
10.10.4.114	hunkTimerSelector	433
10.10.4.115	hunkTimerValue	434
10.10.4.116	hunkTimestamp	434
10.10.4.117	hunkTimestampLatchValue	434
10.10.4.118	hunkTransferBlockID	434
10.10.4.119	hunkTransferQueueCurrentBlockCount	434
10.10.4.120	hunkTransferStreamID	434
10.10.4.121	hunkWidth	434
10.10.4.122	Configuration	435
10.10.4.123	TimeSlotsCount	435
10.10.4.124	ColorTransformationEnable	435
10.10.4.125	ColorTransformationSelector	435
10.10.4.126	ColorTransformationValue	435
10.10.4.127	ColorTransformationValueSelector	435
10.10.4.128	CompressionRatio	436
10.10.4.129	CounterDelay	436
10.10.4.130	CounterDuration	436
10.10.4.131	CounterEventActivation	436

10.10.4.132	CounterEventSource	436
10.10.4.133	CounterReset	436
10.10.4.134	CounterResetActivation	436
10.10.4.135	CounterResetSource	436
10.10.4.136	CounterSelector	437
10.10.4.137	CounterStatus	437
10.10.4.138	CounterTriggerActivation	437
10.10.4.139	CounterTriggerSource	437
10.10.4.140	CounterValue	437
10.10.4.141	CounterValueAtReset	437
10.10.4.142	CxpConnectionSelector	437
10.10.4.143	CxpConnectionTestErrorCount	437
10.10.4.144	CxpConnectionTestMode	438
10.10.4.145	CxpConnectionTestPacketCount	438
10.10.4.146	CxpLinkConfiguration	438
10.10.4.147	CxpLinkConfigurationPreferred	438
10.10.4.148	CxpLinkConfigurationStatus	438
10.10.4.149	CxpPoCxpAuto	438
10.10.4.150	CxpPoCxpStatus	438
10.10.4.151	CxpPoCxpTripReset	439
10.10.4.152	CxpPoCxpTurnOff	439
10.10.4.153	DecimationHorizontal	439
10.10.4.154	DecimationHorizontalMode	439
10.10.4.155	DecimationSelector	439
10.10.4.156	DecimationVertical	440
10.10.4.157	DecimationVerticalMode	440
10.10.4.158	DefectCorrectionMode	440
10.10.4.159	DefectCorrectStaticEnable	440
10.10.4.160	DefectTableApply	440
10.10.4.161	DefectTableCoordinateX	441

10.10.4.162	DefectTableCoordinateY	441
10.10.4.163	DefectTableFactoryRestore	441
10.10.4.164	DefectTableIndex	441
10.10.4.165	DefectTablePixelCount	441
10.10.4.166	DefectTableSave	442
10.10.4.167	Deinterlacing	442
10.10.4.168	DeviceCharacterSet	442
10.10.4.169	DeviceClockFrequency	442
10.10.4.170	DeviceClockSelector	442
10.10.4.171	DeviceConnectionSelector	442
10.10.4.172	DeviceConnectionSpeed	442
10.10.4.173	DeviceConnectionStatus	443
10.10.4.174	DeviceEventChannelCount	443
10.10.4.175	DeviceFamilyName	443
10.10.4.176	DeviceFeaturePersistenceEnd	443
10.10.4.177	DeviceFeaturePersistenceStart	443
10.10.4.178	DeviceFirmwareVersion	443
10.10.4.179	DeviceGenCPVersionMajor	443
10.10.4.180	DeviceGenCPVersionMinor	444
10.10.4.181	DeviceID	444
10.10.4.182	DeviceIndicatorMode	444
10.10.4.183	DeviceLinkBandwidthReserve	444
10.10.4.184	DeviceLinkCommandTimeout	444
10.10.4.185	DeviceLinkConnectionCount	444
10.10.4.186	DeviceLinkCurrentThroughput	444
10.10.4.187	DeviceLinkHeartbeatMode	445
10.10.4.188	DeviceLinkHeartbeatTimeout	445
10.10.4.189	DeviceLinkSelector	445
10.10.4.190	DeviceLinkSpeed	445
10.10.4.191	DeviceLinkThroughputLimit	445

10.10.4.192	DeviceLinkThroughputLimitMode	445
10.10.4.193	DeviceManifestEntrySelector	446
10.10.4.194	DeviceManifestPrimaryURL	446
10.10.4.195	DeviceManifestSchemaMajorVersion	446
10.10.4.196	DeviceManifestSchemaMinorVersion	446
10.10.4.197	DeviceManifestSecondaryURL	446
10.10.4.198	DeviceManifestXMLMajorVersion	446
10.10.4.199	DeviceManifestXMLMinorVersion	446
10.10.4.200	DeviceManifestXMLSubMinorVersion	446
10.10.4.201	DeviceManufacturerInfo	447
10.10.4.202	DeviceMaxThroughput	447
10.10.4.203	DeviceModelName	447
10.10.4.204	DevicePowerSupplySelector	447
10.10.4.205	DeviceRegistersCheck	447
10.10.4.206	DeviceRegistersEndianness	447
10.10.4.207	DeviceRegistersStreamingEnd	448
10.10.4.208	DeviceRegistersStreamingStart	448
10.10.4.209	DeviceRegistersValid	448
10.10.4.210	DeviceReset	448
10.10.4.211	DeviceScanType	448
10.10.4.212	DeviceSerialNumber	448
10.10.4.213	DeviceSerialPortBaudRate	448
10.10.4.214	DeviceSerialPortSelector	449
10.10.4.215	DeviceSFNCVersionMajor	449
10.10.4.216	DeviceSFNCVersionMinor	449
10.10.4.217	DeviceSFNCVersionSubMinor	449
10.10.4.218	DeviceStreamChannelCount	449
10.10.4.219	DeviceStreamChannelEndianness	449
10.10.4.220	DeviceStreamChannelLink	449
10.10.4.221	DeviceStreamChannelPacketSize	450

10.10.4.222	DeviceStreamChannelSelector	450
10.10.4.223	DeviceStreamChannelType	450
10.10.4.224	DeviceTapGeometry	450
10.10.4.225	DeviceTemperature	450
10.10.4.226	DeviceTemperatureSelector	450
10.10.4.227	DeviceTLType	450
10.10.4.228	DeviceTLVersionMajor	451
10.10.4.229	DeviceTLVersionMinor	451
10.10.4.230	DeviceTLVersionSubMinor	451
10.10.4.231	DeviceType	451
10.10.4.232	DeviceUptime	451
10.10.4.233	DeviceUserID	451
10.10.4.234	DeviceVendorName	451
10.10.4.235	DeviceVersion	452
10.10.4.236	EncoderDivider	452
10.10.4.237	EncoderMode	452
10.10.4.238	EncoderOutputMode	452
10.10.4.239	EncoderReset	452
10.10.4.240	EncoderResetActivation	452
10.10.4.241	EncoderResetSource	452
10.10.4.242	EncoderSelector	453
10.10.4.243	EncoderSourceA	453
10.10.4.244	EncoderSourceB	453
10.10.4.245	EncoderStatus	453
10.10.4.246	EncoderTimeout	453
10.10.4.247	EncoderValue	453
10.10.4.248	EncoderValueAtReset	453
10.10.4.249	EnumerationCount	454
10.10.4.250	EventAcquisitionEnd	454
10.10.4.251	EventAcquisitionEndFrameID	454

10.10.4.252	EventAcquisitionEndTimestamp	454
10.10.4.253	EventAcquisitionError	454
10.10.4.254	EventAcquisitionErrorFrameID	454
10.10.4.255	EventAcquisitionErrorTimestamp	454
10.10.4.256	EventAcquisitionStart	454
10.10.4.257	EventAcquisitionStartFrameID	455
10.10.4.258	EventAcquisitionStartTimestamp	455
10.10.4.259	EventAcquisitionTransferEnd	455
10.10.4.260	EventAcquisitionTransferEndFrameID	455
10.10.4.261	EventAcquisitionTransferEndTimestamp	455
10.10.4.262	EventAcquisitionTransferStart	455
10.10.4.263	EventAcquisitionTransferStartFrameID	455
10.10.4.264	EventAcquisitionTransferStartTimestamp	456
10.10.4.265	EventAcquisitionTrigger	456
10.10.4.266	EventAcquisitionTriggerFrameID	456
10.10.4.267	EventAcquisitionTriggerTimestamp	456
10.10.4.268	EventActionLate	456
10.10.4.269	EventActionLateFrameID	456
10.10.4.270	EventActionLateTimestamp	456
10.10.4.271	EventCounter0End	456
10.10.4.272	EventCounter0EndFrameID	457
10.10.4.273	EventCounter0EndTimestamp	457
10.10.4.274	EventCounter0Start	457
10.10.4.275	EventCounter0StartFrameID	457
10.10.4.276	EventCounter0StartTimestamp	457
10.10.4.277	EventCounter1End	457
10.10.4.278	EventCounter1EndFrameID	457
10.10.4.279	EventCounter1EndTimestamp	457
10.10.4.280	EventCounter1Start	458
10.10.4.281	EventCounter1StartFrameID	458

10.10.4.282	EventCounter1StartTimestamp	458
10.10.4.283	EventEncoder0Restarted	458
10.10.4.284	EventEncoder0RestartedFrameID	458
10.10.4.285	EventEncoder0RestartedTimestamp	458
10.10.4.286	EventEncoder0Stopped	458
10.10.4.287	EventEncoder0StoppedFrameID	458
10.10.4.288	EventEncoder0StoppedTimestamp	459
10.10.4.289	EventEncoder1Restarted	459
10.10.4.290	EventEncoder1RestartedFrameID	459
10.10.4.291	EventEncoder1RestartedTimestamp	459
10.10.4.292	EventEncoder1Stopped	459
10.10.4.293	EventEncoder1StoppedFrameID	459
10.10.4.294	EventEncoder1StoppedTimestamp	459
10.10.4.295	EventError	459
10.10.4.296	EventErrorCode	460
10.10.4.297	EventErrorFrameID	460
10.10.4.298	EventErrorTimestamp	460
10.10.4.299	EventExposureEnd	460
10.10.4.300	EventExposureEndFrameID	460
10.10.4.301	EventExposureEndTimestamp	460
10.10.4.302	EventExposureStart	460
10.10.4.303	EventExposureStartFrameID	460
10.10.4.304	EventExposureStartTimestamp	461
10.10.4.305	EventFrameBurstEnd	461
10.10.4.306	EventFrameBurstEndFrameID	461
10.10.4.307	EventFrameBurstEndTimestamp	461
10.10.4.308	EventFrameBurstStart	461
10.10.4.309	EventFrameBurstStartFrameID	461
10.10.4.310	EventFrameBurstStartTimestamp	461
10.10.4.311	EventFrameEnd	461



10.10.4.312	EventFrameEndFrameID	462
10.10.4.313	EventFrameEndTimestamp	462
10.10.4.314	EventFrameStart	462
10.10.4.315	EventFrameStartFrameID	462
10.10.4.316	EventFrameStartTimestamp	462
10.10.4.317	EventFrameTransferEnd	462
10.10.4.318	EventFrameTransferEndFrameID	462
10.10.4.319	EventFrameTransferEndTimestamp	462
10.10.4.320	EventFrameTransferStart	463
10.10.4.321	EventFrameTransferStartFrameID	463
10.10.4.322	EventFrameTransferStartTimestamp	463
10.10.4.323	EventFrameTrigger	463
10.10.4.324	EventFrameTriggerFrameID	463
10.10.4.325	EventFrameTriggerTimestamp	463
10.10.4.326	EventLine0AnyEdge	463
10.10.4.327	EventLine0AnyEdgeFrameID	464
10.10.4.328	EventLine0AnyEdgeTimestamp	464
10.10.4.329	EventLine0FallingEdge	464
10.10.4.330	EventLine0FallingEdgeFrameID	464
10.10.4.331	EventLine0FallingEdgeTimestamp	464
10.10.4.332	EventLine0RisingEdge	464
10.10.4.333	EventLine0RisingEdgeFrameID	464
10.10.4.334	EventLine0RisingEdgeTimestamp	464
10.10.4.335	EventLine1AnyEdge	465
10.10.4.336	EventLine1AnyEdgeFrameID	465
10.10.4.337	EventLine1AnyEdgeTimestamp	465
10.10.4.338	EventLine1FallingEdge	465
10.10.4.339	EventLine1FallingEdgeFrameID	465
10.10.4.340	EventLine1FallingEdgeTimestamp	465
10.10.4.341	EventLine1RisingEdge	465

10.10.4.342	EventLine1RisingEdgeFrameID	465
10.10.4.343	EventLine1RisingEdgeTimestamp	466
10.10.4.344	EventLinkSpeedChange	466
10.10.4.345	EventLinkSpeedChangeFrameID	466
10.10.4.346	EventLinkSpeedChangeTimestamp	466
10.10.4.347	EventLinkTrigger0	466
10.10.4.348	EventLinkTrigger0FrameID	466
10.10.4.349	EventLinkTrigger0Timestamp	466
10.10.4.350	EventLinkTrigger1	466
10.10.4.351	EventLinkTrigger1FrameID	467
10.10.4.352	EventLinkTrigger1Timestamp	467
10.10.4.353	EventNotification	467
10.10.4.354	EventSelector	467
10.10.4.355	EventSequencerSetChange	467
10.10.4.356	EventSequencerSetChangeFrameID	467
10.10.4.357	EventSequencerSetChangeTimestamp	467
10.10.4.358	EventSerialData	467
10.10.4.359	EventSerialDataLength	468
10.10.4.360	EventSerialPortReceive	468
10.10.4.361	EventSerialPortReceiveTimestamp	468
10.10.4.362	EventSerialReceiveOverflow	468
10.10.4.363	EventStream0TransferBlockEnd	468
10.10.4.364	EventStream0TransferBlockEndFrameID	468
10.10.4.365	EventStream0TransferBlockEndTimestamp	468
10.10.4.366	EventStream0TransferBlockStart	469
10.10.4.367	EventStream0TransferBlockStartFrameID	469
10.10.4.368	EventStream0TransferBlockStartTimestamp	469
10.10.4.369	EventStream0TransferBlockTrigger	469
10.10.4.370	EventStream0TransferBlockTriggerFrameID	469
10.10.4.371	EventStream0TransferBlockTriggerTimestamp	469

10.10.4.372	EventStream0TransferBurstEnd	469
10.10.4.373	EventStream0TransferBurstEndFrameID	470
10.10.4.374	EventStream0TransferBurstEndTimestamp	470
10.10.4.375	EventStream0TransferBurstStart	470
10.10.4.376	EventStream0TransferBurstStartFrameID	470
10.10.4.377	EventStream0TransferBurstStartTimestamp	470
10.10.4.378	EventStream0TransferEnd	470
10.10.4.379	EventStream0TransferEndFrameID	470
10.10.4.380	EventStream0TransferEndTimestamp	471
10.10.4.381	EventStream0TransferOverflow	471
10.10.4.382	EventStream0TransferOverflowFrameID	471
10.10.4.383	EventStream0TransferOverflowTimestamp	471
10.10.4.384	EventStream0TransferPause	471
10.10.4.385	EventStream0TransferPauseFrameID	471
10.10.4.386	EventStream0TransferPauseTimestamp	471
10.10.4.387	EventStream0TransferResume	472
10.10.4.388	EventStream0TransferResumeFrameID	472
10.10.4.389	EventStream0TransferResumeTimestamp	472
10.10.4.390	EventStream0TransferStart	472
10.10.4.391	EventStream0TransferStartFrameID	472
10.10.4.392	EventStream0TransferStartTimestamp	472
10.10.4.393	EventTest	472
10.10.4.394	EventTestTimestamp	473
10.10.4.395	EventTimer0End	473
10.10.4.396	EventTimer0EndFrameID	473
10.10.4.397	EventTimer0EndTimestamp	473
10.10.4.398	EventTimer0Start	473
10.10.4.399	EventTimer0StartFrameID	473
10.10.4.400	EventTimer0StartTimestamp	473
10.10.4.401	EventTimer1End	473

10.10.4.402	EventTimer1EndFrameID	474
10.10.4.403	EventTimer1EndTimestamp	474
10.10.4.404	EventTimer1Start	474
10.10.4.405	EventTimer1StartFrameID	474
10.10.4.406	EventTimer1StartTimestamp	474
10.10.4.407	ExposureActiveMode	474
10.10.4.408	ExposureAuto	474
10.10.4.409	ExposureMode	474
10.10.4.410	ExposureTime	475
10.10.4.411	ExposureTimeMode	475
10.10.4.412	ExposureTimeSelector	475
10.10.4.413	FactoryReset	475
10.10.4.414	FileAccessBuffer	475
10.10.4.415	FileAccessLength	475
10.10.4.416	FileAccessOffset	475
10.10.4.417	FileOpenMode	476
10.10.4.418	FileOperationExecute	476
10.10.4.419	FileOperationResult	476
10.10.4.420	FileOperationSelector	476
10.10.4.421	FileOperationStatus	476
10.10.4.422	FileSelector	476
10.10.4.423	FileSize	477
10.10.4.424	Gain	477
10.10.4.425	GainAuto	477
10.10.4.426	GainAutoBalance	477
10.10.4.427	GainSelector	477
10.10.4.428	Gamma	477
10.10.4.429	GammaEnable	477
10.10.4.430	GenActiveLinkCount	478
10.10.4.431	GenCCP	478

10.10.4.432	DevCurrentDefaultGateway	478
10.10.4.433	DevCurrentIPAddress	478
10.10.4.434	DevCurrentIPConfigurationDHCP	478
10.10.4.435	DevCurrentIPConfigurationLLA	478
10.10.4.436	DevCurrentIPConfigurationPersistentIP	478
10.10.4.437	DevCurrentPhysicalLinkConfiguration	478
10.10.4.438	DevCurrentSubnetMask	479
10.10.4.439	DevDiscoveryAckDelay	479
10.10.4.440	DevFirstURL	479
10.10.4.441	DevGVCPExtendedStatusCodes	479
10.10.4.442	DevGVCPExtendedStatusCodesSelector	479
10.10.4.443	DevGVCPHeartbeatDisable	479
10.10.4.444	DevGVCPPendingAck	479
10.10.4.445	DevGVCPPendingTimeout	480
10.10.4.446	DevGVSPExtendedIDMode	480
10.10.4.447	DevHeartbeatTimeout	480
10.10.4.448	DevIEEE1588	480
10.10.4.449	DevIEEE1588ClockAccuracy	480
10.10.4.450	DevIEEE1588Mode	480
10.10.4.451	DevIEEE1588Status	480
10.10.4.452	DevInterfaceSelector	481
10.10.4.453	DevIPConfigurationStatus	481
10.10.4.454	DevMACAddress	481
10.10.4.455	DevMCDA	481
10.10.4.456	DevMCPHostPort	481
10.10.4.457	DevMCRC	481
10.10.4.458	DevMCSP	481
10.10.4.459	DevMCTT	481
10.10.4.460	DevNumberOfInterfaces	482
10.10.4.461	DevPAUSEFrameReception	482

10.10.4.462	DevPAUSEFrameTransmission	482
10.10.4.463	DevPersistentDefaultGateway	482
10.10.4.464	DevPersistentIPAddress	482
10.10.4.465	DevPersistentSubnetMask	482
10.10.4.466	DevPhysicalLinkConfiguration	482
10.10.4.467	DevPrimaryApplicationIPAddress	482
10.10.4.468	DevPrimaryApplicationSocket	483
10.10.4.469	DevPrimaryApplicationSwitchoverKey	483
10.10.4.470	DevSCCFGAllInTransmission	483
10.10.4.471	DevSCCFGExtendedChunkData	483
10.10.4.472	DevSCCFGPacketResendDestination	483
10.10.4.473	DevSCCFGUnconditionalStreaming	483
10.10.4.474	DevSCDA	483
10.10.4.475	DevSCPD	484
10.10.4.476	DevSCPDDirection	484
10.10.4.477	DevSCPHostPort	484
10.10.4.478	DevSCPIInterfaceIndex	484
10.10.4.479	DevSCPSBigEndian	484
10.10.4.480	DevSCPSDoNotFragment	484
10.10.4.481	DevSCPSFireTestPacket	484
10.10.4.482	DevSCPSPacketSize	485
10.10.4.483	DevSCSP	485
10.10.4.484	DevSCZoneConfigurationLock	485
10.10.4.485	DevSCZoneCount	485
10.10.4.486	DevSCZoneDirectionAll	485
10.10.4.487	DevSecondURL	485
10.10.4.488	DevStreamChannelSelector	485
10.10.4.489	DevSupportedOption	486
10.10.4.490	DevSupportedOptionSelector	486
10.10.4.491	DevTimestampTickFrequency	486

10.10.4.492	<a href="#">XmlManifestAddress</a>	486
10.10.4.493	<a href="#">Height</a>	486
10.10.4.494	<a href="#">HeightMax</a>	486
10.10.4.495	<a href="#">ImageComponentEnable</a>	486
10.10.4.496	<a href="#">ImageComponentSelector</a>	487
10.10.4.497	<a href="#">ImageCompressionBitrate</a>	487
10.10.4.498	<a href="#">ImageCompressionJPEGFormatOption</a>	487
10.10.4.499	<a href="#">ImageCompressionMode</a>	487
10.10.4.500	<a href="#">ImageCompressionQuality</a>	487
10.10.4.501	<a href="#">ImageCompressionRateOption</a>	487
10.10.4.502	<a href="#">IspEnable</a>	487
10.10.4.503	<a href="#">IspFilterWidth</a>	488
10.10.4.504	<a href="#">IspFormat</a>	488
10.10.4.505	<a href="#">IspInputFilterSelector</a>	488
10.10.4.506	<a href="#">IspInverter</a>	488
10.10.4.507	<a href="#">IspMode</a>	488
10.10.4.508	<a href="#">IspPitch</a>	488
10.10.4.509	<a href="#">IspSelector</a>	488
10.10.4.510	<a href="#">IspSource</a>	488
10.10.4.511	<a href="#">IspStatus</a>	489
10.10.4.512	<a href="#">IspStatusAll</a>	489
10.10.4.513	<a href="#">IspErrorCount</a>	489
10.10.4.514	<a href="#">IspUptime</a>	489
10.10.4.515	<a href="#">LogicBlockLUTInputActivation</a>	489
10.10.4.516	<a href="#">LogicBlockLUTInputSelector</a>	489
10.10.4.517	<a href="#">LogicBlockLUTInputSource</a>	489
10.10.4.518	<a href="#">LogicBlockLUTOutputValue</a>	489
10.10.4.519	<a href="#">LogicBlockLUTOutputValueAll</a>	490
10.10.4.520	<a href="#">LogicBlockLUTRowIndex</a>	490
10.10.4.521	<a href="#">LogicBlockLUTSelector</a>	490

10.10.4.522	LogicBlockSelector	490
10.10.4.523	UTEnable	490
10.10.4.524	UTIndex	490
10.10.4.525	UTSelector	490
10.10.4.526	UTValue	491
10.10.4.527	UTValueAll	491
10.10.4.528	MaxDeviceResetTime	491
10.10.4.529	OffsetX	491
10.10.4.530	OffsetY	491
10.10.4.531	PacketResendRequestCount	491
10.10.4.532	PayloadSize	491
10.10.4.533	PixelColorFilter	492
10.10.4.534	PixelDynamicRangeMax	492
10.10.4.535	PixelDynamicRangeMin	492
10.10.4.536	PixelFormat	492
10.10.4.537	PixelFormatInfoID	492
10.10.4.538	PixelFormatInfoSelector	492
10.10.4.539	PixelSize	492
10.10.4.540	PowerSupplyCurrent	493
10.10.4.541	PowerSupplyVoltage	493
10.10.4.542	RegionDestination	493
10.10.4.543	RegionMode	493
10.10.4.544	RegionSelector	493
10.10.4.545	ReverseX	493
10.10.4.546	ReverseY	494
10.10.4.547	RgbTransformLightSource	494
10.10.4.548	Saturation	494
10.10.4.549	SaturationEnable	494
10.10.4.550	Scan3dAxisMax	494
10.10.4.551	Scan3dAxisMin	494



10.10.4.552	Scan3dCoordinateOffset	495
10.10.4.553	Scan3dCoordinateReferenceSelector	495
10.10.4.554	Scan3dCoordinateReferenceValue	495
10.10.4.555	Scan3dCoordinateScale	495
10.10.4.556	Scan3dCoordinateSelector	495
10.10.4.557	Scan3dCoordinateSystem	495
10.10.4.558	Scan3dCoordinateSystemReference	495
10.10.4.559	Scan3dCoordinateTransformSelector	496
10.10.4.560	Scan3dDistanceUnit	496
10.10.4.561	Scan3dInvalidDataFlag	496
10.10.4.562	Scan3dInvalidDataValue	496
10.10.4.563	Scan3dOutputMode	496
10.10.4.564	Scan3dTransformValue	496
10.10.4.565	SensorDescription	496
10.10.4.566	SensorDigitizationTaps	497
10.10.4.567	SensorHeight	497
10.10.4.568	SensorShutterMode	497
10.10.4.569	SensorTaps	497
10.10.4.570	SensorWidth	497
10.10.4.571	SequencerConfigurationMode	497
10.10.4.572	SequencerConfigurationValid	497
10.10.4.573	SequencerFeatureEnable	498
10.10.4.574	SequencerMode	498
10.10.4.575	SequencerPathSelector	498
10.10.4.576	SequencerSetActive	498
10.10.4.577	SequencerSetLoad	498
10.10.4.578	SequencerSetNext	498
10.10.4.579	SequencerSetSave	499
10.10.4.580	SequencerSetSelector	499
10.10.4.581	SequencerSetStart	499

10.10.4.582	SequencerSetValid . . . . .	499
10.10.4.583	SequencerTriggerActivation . . . . .	499
10.10.4.584	SequencerTriggerSource . . . . .	499
10.10.4.585	SerialPortBaudRate . . . . .	500
10.10.4.586	SerialPortDataBits . . . . .	500
10.10.4.587	SerialPortParity . . . . .	500
10.10.4.588	SerialPortSelector . . . . .	500
10.10.4.589	SerialPortSource . . . . .	500
10.10.4.590	SerialPortStopBits . . . . .	500
10.10.4.593	SerialReceiveFramingErrorCount . . . . .	500
10.10.4.594	SerialReceiveParityErrorCount . . . . .	500
10.10.4.595	SerialReceiveQueueClear . . . . .	501
10.10.4.596	SerialReceiveQueueCurrentCharacterCount . . . . .	501
10.10.4.597	SerialReceiveQueueMaxCharacterCount . . . . .	501
10.10.4.598	SerialTransmitQueueCurrentCharacterCount . . . . .	501
10.10.4.599	SerialTransmitQueueMaxCharacterCount . . . . .	501
10.10.4.600	Sharpening . . . . .	501
10.10.4.601	SharpeningAuto . . . . .	502
10.10.4.602	SharpeningEnable . . . . .	502
10.10.4.603	SharpeningThreshold . . . . .	502
10.10.4.604	SoftwareSignalPulse . . . . .	502
10.10.4.605	SoftwareSignalSelector . . . . .	502
10.10.4.606	SourceCount . . . . .	503
10.10.4.607	SourceSelector . . . . .	503
10.10.4.608	Test0001 . . . . .	503
10.10.4.609	TestEventGenerate . . . . .	503
10.10.4.610	TestPattern . . . . .	503
10.10.4.611	TestPatternGeneratorSelector . . . . .	503
10.10.4.612	TestPendingAck . . . . .	503
10.10.4.613	TimerDelay . . . . .	504

10.10.4.61TimerDuration . . . . .	504
10.10.4.61TimerReset . . . . .	504
10.10.4.61TimerSelector . . . . .	504
10.10.4.61TimerStatus . . . . .	504
10.10.4.61TimerTriggerActivation . . . . .	504
10.10.4.61TimerTriggerSource . . . . .	504
10.10.4.61TimerValue . . . . .	505
10.10.4.61Timestamp . . . . .	505
10.10.4.62TimestampLatch . . . . .	505
10.10.4.62TimestampLatchValue . . . . .	505
10.10.4.62TimestampReset . . . . .	505
10.10.4.62BLParamsLocked . . . . .	505
10.10.4.62TransferAbort . . . . .	505
10.10.4.62TransferBlockCount . . . . .	506
10.10.4.62TransferBurstCount . . . . .	506
10.10.4.62TransferComponentSelector . . . . .	506
10.10.4.62TransferControlMode . . . . .	506
10.10.4.62TransferOperationMode . . . . .	506
10.10.4.63TransferPause . . . . .	506
10.10.4.63TransferQueueCurrentBlockCount . . . . .	506
10.10.4.63TransferQueueMaxBlockCount . . . . .	507
10.10.4.63TransferQueueMode . . . . .	507
10.10.4.63TransferQueueOverflowCount . . . . .	507
10.10.4.63TransferResume . . . . .	507
10.10.4.63TransferSelector . . . . .	507
10.10.4.63TransferStart . . . . .	507
10.10.4.63TransferStatus . . . . .	507
10.10.4.63TransferStatusSelector . . . . .	507
10.10.4.64TransferStop . . . . .	508
10.10.4.64TransferStreamChannel . . . . .	508

10.10.4.64TransferTriggerActivation . . . . .	508
10.10.4.64TransferTriggerMode . . . . .	508
10.10.4.64TransferTriggerSelector . . . . .	508
10.10.4.64TransferTriggerSource . . . . .	508
10.10.4.64TriggerActivation . . . . .	508
10.10.4.64TriggerDelay . . . . .	509
10.10.4.64TriggerDivider . . . . .	509
10.10.4.64TriggerEventTest . . . . .	509
10.10.4.65TriggerMode . . . . .	509
10.10.4.65TriggerMultiplier . . . . .	509
10.10.4.65TriggerOverlap . . . . .	509
10.10.4.65TriggerSelector . . . . .	509
10.10.4.65TriggerSoftware . . . . .	510
10.10.4.65TriggerSource . . . . .	510
10.10.4.65UserOutputSelector . . . . .	510
10.10.4.65UserOutputValue . . . . .	510
10.10.4.65UserOutputValueAll . . . . .	510
10.10.4.65UserOutputValueAllMask . . . . .	510
10.10.4.66UserSetDefault . . . . .	511
10.10.4.66UserSetFeatureEnable . . . . .	511
10.10.4.66UserSetLoad . . . . .	511
10.10.4.66UserSetSave . . . . .	511
10.10.4.66UserSetSelector . . . . .	511
10.10.4.66UserSet3_Enable . . . . .	511
10.10.4.66WhiteClip . . . . .	512
10.10.4.66WhiteClipSelector . . . . .	512
10.10.4.66Width . . . . .	512
10.10.4.66WidthMax . . . . .	512
10.11 CameraBase Class Reference . . . . .	513
10.11.1 Detailed Description . . . . .	515

10.11.2 Constructor & Destructor Documentation . . . . .	515
10.11.2.1 ~CameraBase(void) . . . . .	515
10.11.2.2 CameraBase(void) . . . . .	515
10.11.2.3 CameraBase(const CameraBase &) . . . . .	515
10.11.3 Member Function Documentation . . . . .	515
10.11.3.1 BeginAcquisition() . . . . .	515
10.11.3.2 Delnit() . . . . .	516
10.11.3.3 DiscoverMaxPacketSize() . . . . .	516
10.11.3.4 EndAcquisition() . . . . .	516
10.11.3.5 GetAccessMode() const . . . . .	516
10.11.3.6 GetGuiXml() const . . . . .	517
10.11.3.7 GetNextImage(uint64_t grabTimeout=EVENT_TIMEOUT_INFINITE, uint64_t streamID=0) . . . . .	517
10.11.3.8 GetNodeMap() const . . . . .	517
10.11.3.9 GetNumDataStreams() . . . . .	518
10.11.3.10 GetNumImagesInUse() . . . . .	518
10.11.3.11 GetTLDeviceNodeMap() const . . . . .	518
10.11.3.12 GetTLStreamNodeMap() const . . . . .	518
10.11.3.13 GetUniqueID() . . . . .	519
10.11.3.14 Init() . . . . .	519
10.11.3.15 IsInitialized() . . . . .	519
10.11.3.16 IsStreaming() const . . . . .	520
10.11.3.17 IsValid() . . . . .	520
10.11.3.18 operator=(const CameraBase &) . . . . .	520
10.11.3.19 ReadPort(uint64_t iAddress, void *pBuffer, size_t iSize) . . . . .	520
10.11.3.20 RegisterEvent(Event &evtToRegister) . . . . .	520
10.11.3.21 RegisterEvent(Event &evtToRegister, const GenICam::gcstring &eventName) . . . . .	521
10.11.3.22 UnregisterEvent(Event &evtToUnregister) . . . . .	521
10.11.3.23 WritePort(uint64_t iAddress, const void *pBuffer, size_t iSize) . . . . .	521
10.11.4 Friends And Related Function Documentation . . . . .	521
10.11.4.1 InterfaceImpl . . . . .	521

10.12	CameraList Class Reference	522
10.12.1	Detailed Description	523
10.12.2	Constructor & Destructor Documentation	523
10.12.2.1	CameraList(void)	523
10.12.2.2	~CameraList(void)	523
10.12.2.3	CameraList(const CameraList &iface)	523
10.12.3	Member Function Documentation	523
10.12.3.1	Append(CameraList &otherList)	523
10.12.3.2	Clear()	524
10.12.3.3	GetByIndex(unsigned int index) const	524
10.12.3.4	GetBySerial(std::string serialNumber) const	524
10.12.3.5	GetSize() const	525
10.12.3.6	operator=(const CameraList &iface)	525
10.12.3.7	operator[](unsigned int index)	525
10.12.3.8	RemoveByIndex(unsigned int index)	525
10.12.3.9	RemoveBySerial(std::string serialNumber)	525
10.13	CameraPtr Class Reference	526
10.13.1	Detailed Description	527
10.14	CategoryNode Class Reference	527
10.14.1	Detailed Description	528
10.14.2	Constructor & Destructor Documentation	528
10.14.2.1	CategoryNode()	528
10.14.2.2	CategoryNode(std::shared_ptr< Node::NodeImpl > pCategory)	528
10.14.2.3	~CategoryNode()	528
10.14.3	Member Function Documentation	528
10.14.3.1	GetFeatures(FeatureList_t &Features) const	528
10.14.3.2	SetReference(INode *pBase)	529
10.15	CChunkAdapter Class Reference	529
10.15.1	Detailed Description	530
10.15.2	Constructor & Destructor Documentation	530

10.15.2.1 ~CChunkAdapter()	530
10.15.2.2 CChunkAdapter(INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)	530
10.15.3 Member Function Documentation	530
10.15.3.1 AttachBuffer(uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL)=0	530
10.15.3.2 AttachNodeMap(INodeMap *pNodeMap)	530
10.15.3.3 CheckBufferLayout(uint8_t *pBuffer, int64_t BufferLength)=0	530
10.15.3.4 ClearCaches()	530
10.15.3.5 DetachBuffer()	531
10.15.3.6 DetachNodeMap()	531
10.15.3.7 UpdateBuffer(uint8_t *pBaseAddress)	531
10.15.4 Member Data Documentation	531
10.15.4.1 m_pChunkAdapter	531
10.16 CChunkAdapterDcam Class Reference	531
10.16.1 Detailed Description	532
10.16.2 Constructor & Destructor Documentation	532
10.16.2.1 CChunkAdapterDcam(INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)	532
10.16.2.2 ~CChunkAdapterDcam()	532
10.16.3 Member Function Documentation	533
10.16.3.1 AttachBuffer(uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL)	533
10.16.3.2 CheckBufferLayout(uint8_t *pBuffer, int64_t BufferLength)	533
10.16.3.3 CheckCRC(uint8_t *pBuffer, int64_t BufferLength)	533
10.16.3.4 HasCRC(uint8_t *pBuffer, int64_t BufferLength)	533
10.17 CChunkAdapterGeneric Class Reference	533
10.17.1 Constructor & Destructor Documentation	534
10.17.1.1 CChunkAdapterGeneric(INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)	534
10.17.1.2 ~CChunkAdapterGeneric()	534
10.17.2 Member Function Documentation	534
10.17.2.1 AttachBuffer(uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL)	534

10.17.2.2 AttachBuffer(uint8_t *pBuffer, SingleChunkData_t *ChunkData, int64_t NumChunks, AttachStatistics_t *pAttachStatistics=NULL) . . . . .	535
10.17.2.3 AttachBuffer(uint8_t *pBuffer, SingleChunkDataStr_t *ChunkData, int64_t NumChunks, AttachStatistics_t *pAttachStatistics=NULL) . . . . .	535
10.17.2.4 CheckBufferLayout(uint8_t *pBuffer, int64_t BufferLength) . . . . .	535
10.18 CChunkAdapterGEV Class Reference . . . . .	535
10.18.1 Detailed Description . . . . .	536
10.18.2 Constructor & Destructor Documentation . . . . .	536
10.18.2.1 CChunkAdapterGEV(INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1) . . . . .	536
10.18.2.2 ~CChunkAdapterGEV() . . . . .	536
10.18.3 Member Function Documentation . . . . .	536
10.18.3.1 AttachBuffer(uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL) . . . . .	536
10.18.3.2 CheckBufferLayout(uint8_t *pBuffer, int64_t BufferLength) . . . . .	536
10.19 CChunkAdapterU3V Class Reference . . . . .	537
10.19.1 Detailed Description . . . . .	537
10.19.2 Constructor & Destructor Documentation . . . . .	538
10.19.2.1 CChunkAdapterU3V(INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1) . . . . .	538
10.19.2.2 ~CChunkAdapterU3V() . . . . .	538
10.19.3 Member Function Documentation . . . . .	538
10.19.3.1 AttachBuffer(uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL) . . . . .	538
10.19.3.2 CheckBufferLayout(uint8_t *pBuffer, int64_t BufferLength) . . . . .	538
10.20 CChunkPort Class Reference . . . . .	538
10.20.1 Detailed Description . . . . .	540
10.20.2 Constructor & Destructor Documentation . . . . .	540
10.20.2.1 CChunkPort(IPort *pPort=NULL) . . . . .	540
10.20.2.2 ~CChunkPort() . . . . .	540
10.20.3 Member Function Documentation . . . . .	540
10.20.3.1 AttachChunk(uint8_t *pBaseAddress, int64_t ChunkOffset, int64_t Length, bool Cache) . . . . .	540



10.20.3.2 AttachPort(::Spinnaker::GenApi::IPort *pPort) . . . . .	540
10.20.3.3 CheckChunkID(uint8_t *pChunkIDBuffer, int ChunkIDLength) . . . . .	540
10.20.3.4 CheckChunkID(uint64_t ChunkID) . . . . .	540
10.20.3.5 ClearCache() . . . . .	540
10.20.3.6 DetachChunk() . . . . .	541
10.20.3.7 DetachPort() . . . . .	541
10.20.3.8 GetAccessMode() const . . . . .	541
10.20.3.9 GetChunkIDLength() . . . . .	541
10.20.3.10 GetPrincipalInterfaceType() const . . . . .	541
10.20.3.11 GetSwapEndianness() . . . . .	541
10.20.3.12 InvalidateNode() . . . . .	541
10.20.3.13 Read(void *pBuffer, int64_t Address, int64_t Length) . . . . .	541
10.20.3.14 SetPortImpl(IPort *pPort) . . . . .	541
10.20.3.15 UpdateBuffer(uint8_t *pBaseAddress) . . . . .	541
10.20.3.16 Write(const void *pBuffer, int64_t Address, int64_t Length) . . . . .	541
10.20.4 Member Data Documentation . . . . .	542
10.20.4.1 m_pChunkPort . . . . .	542
10.20.4.2 m_pPort . . . . .	542
10.20.4.3 m_pPortAdapter . . . . .	542
10.21 CEnumerationTRef< EnumT > Class Template Reference . . . . .	542
10.21.1 Detailed Description . . . . .	544
10.21.2 Constructor & Destructor Documentation . . . . .	544
10.21.2.1 CEnumerationTRef() . . . . .	544
10.21.2.2 CEnumerationTRef(std::shared_ptr< Node::NodeImpl > pEnumeration) . . . . .	544
10.21.2.3 ~CEnumerationTRef() . . . . .	544
10.21.3 Member Function Documentation . . . . .	544
10.21.3.1 GetCurrentEntry(bool Verify=false, bool IgnoreCache=false) . . . . .	544
10.21.3.2 GetEntry(const EnumT Value) . . . . .	544
10.21.3.3 GetEntry(const int64_t IntValue) . . . . .	544
10.21.3.4 GetValue(bool Verify=false, bool IgnoreCache=false) . . . . .	544

10.21.3.5 operator>()	545
10.21.3.6 operator=(EnumT Value)	545
10.21.3.7 operator=(const GenICam::gcstring &ValueStr)	545
10.21.3.8 SetEnumReference(int Index, GenICam::gcstring Name)	545
10.21.3.9 SetNumEnums(int NumEnums)	545
10.21.3.10 SetReference(INode *pBase)	545
10.21.3.11 SetValue(EnumT Value, bool Verify=true)	545
10.22 CEventAdapter Class Reference	546
10.22.1 Detailed Description	546
10.22.2 Constructor & Destructor Documentation	547
10.22.2.1 CEventAdapter(INodeMap *pNodeMap=NULL)	547
10.22.2.2 ~CEventAdapter()	547
10.22.3 Member Function Documentation	547
10.22.3.1 AttachNodeMap(INodeMap *pNodeMap)	547
10.22.3.2 DeliverMessage(const uint8_t msg[], uint32_t numBytes)=0	547
10.22.3.3 DetachNodeMap()	547
10.22.4 Member Data Documentation	547
10.22.4.1 m_pEventAdapter	547
10.23 CEventAdapter1394 Class Reference	548
10.23.1 Detailed Description	548
10.23.2 Constructor & Destructor Documentation	549
10.23.2.1 CEventAdapter1394(INodeMap *pNodeMap=NULL)	549
10.23.2.2 ~CEventAdapter1394()	549
10.23.3 Member Function Documentation	549
10.23.3.1 DeliverEventMessage(EventData1394 &Event, uint32_t numBytes)	549
10.23.3.2 DeliverMessage(const uint8_t msg[], uint32_t numBytes)	549
10.24 CEventAdapterGeneric Class Reference	549
10.24.1 Detailed Description	550
10.24.2 Constructor & Destructor Documentation	550
10.24.2.1 CEventAdapterGeneric(INodeMap *pNodeMap=NULL)	550

10.24.2.2 ~CEventAdapterGeneric()	550
10.24.3 Member Function Documentation	551
10.24.3.1 DeliverMessage(const uint8_t msg[], uint32_t numBytes)	551
10.24.3.2 DeliverMessage(const uint8_t msg[], uint32_t numBytes, const GenICam↵ ::gcstring &EventID)	551
10.24.3.3 DeliverMessage(const uint8_t msg[], uint32_t numBytes, uint64_t EventID)	551
10.25 CEventAdapterGEV Class Reference	551
10.25.1 Detailed Description	552
10.25.2 Constructor & Destructor Documentation	552
10.25.2.1 CEventAdapterGEV(INodeMap *pNodeMap=NULL)	552
10.25.2.2 ~CEventAdapterGEV()	552
10.25.3 Member Function Documentation	552
10.25.3.1 DeliverEventMessage(const GVCP_EVENT_REQUEST *pEvent)	552
10.25.3.2 DeliverEventMessage(const GVCP_EVENTDATA_REQUEST *pEventData)	552
10.25.3.3 DeliverMessage(const uint8_t msg[], uint32_t numBytes)	552
10.26 CEventAdapterU3V Class Reference	553
10.26.1 Detailed Description	553
10.26.2 Constructor & Destructor Documentation	554
10.26.2.1 CEventAdapterU3V(INodeMap *pNodeMap=NULL)	554
10.26.2.2 ~CEventAdapterU3V()	554
10.26.3 Member Function Documentation	554
10.26.3.1 DeliverEventMessage(const U3V_EVENT_MESSAGE *pEventMessage)	554
10.26.3.2 DeliverMessage(const uint8_t msg[], uint32_t numBytes)	554
10.27 CEventPort Class Reference	554
10.27.1 Detailed Description	556
10.27.2 Constructor & Destructor Documentation	556
10.27.2.1 CEventPort(INode *pNode=NULL)	556
10.27.2.2 ~CEventPort()	556
10.27.3 Member Function Documentation	556
10.27.3.1 AttachEvent(uint8_t *pBaseAddress, int64_t Length)	556
10.27.3.2 AttachNode(::Spinnaker::GenApi::INode *pNode)	556

10.27.3.3 CheckEventID(uint8_t *pEventIDBuffer, int EventIDLength)	556
10.27.3.4 CheckEventID(uint64_t EventID)	556
10.27.3.5 DetachEvent()	556
10.27.3.6 DetachNode()	557
10.27.3.7 GetAccessMode() const	557
10.27.3.8 GetEventIDLength()	557
10.27.3.9 GetPrincipalInterfaceType() const	557
10.27.3.10 GetSwapEndianness()	557
10.27.3.11 InvalidateNode()	557
10.27.3.12 Read(void *pBuffer, int64_t Address, int64_t Length)	557
10.27.3.13 SetPortImpl(::Spinnaker::GenApi::IPort *pPort)	557
10.27.3.14 Write(const void *pBuffer, int64_t Address, int64_t Length)	557
10.27.4 Member Data Documentation	557
10.27.4.1 m_pEventPort	557
10.27.4.2 m_pNode	557
10.27.4.3 m_pPortAdapter	557
10.28 CFeatureBag Class Reference	558
10.28.1 Detailed Description	559
10.28.2 Constructor & Destructor Documentation	559
10.28.2.1 CFeatureBag()	559
10.28.2.2 ~CFeatureBag()	559
10.28.3 Member Function Documentation	559
10.28.3.1 GetFeatureBagHandle()	559
10.28.3.2 LoadFromBag(INodeMap *pNodeMap, bool Verify=true, GenICam::gcstring_← vector *pErrorList=NULL)	559
10.28.3.3 operator==(const CFeatureBag &FeatureBag) const	559
10.28.3.4 PersistFeature(IValue &item)	559
10.28.3.5 SetInfo(GenICam::gcstring &Info)	559
10.28.3.6 StoreToBag(INodeMap *pNodeMap, const int MaxNumPersistSkriptEntries=-1)	559
10.29 CFloatPtr Class Reference	560
10.29.1 Detailed Description	561

10.29.2 Constructor & Destructor Documentation . . . . .	561
10.29.2.1 CFloatPtr() . . . . .	561
10.29.2.2 CFloatPtr(IBase *pB) . . . . .	561
10.29.3 Member Function Documentation . . . . .	561
10.29.3.1 GetEnumAlias() . . . . .	561
10.29.3.2 GetIntAlias() . . . . .	561
10.29.3.3 operator=(IBase *pB) . . . . .	561
10.30 CGeneric_XMLLoaderParams Class Reference . . . . .	562
10.30.1 Detailed Description . . . . .	562
10.30.2 Member Function Documentation . . . . .	562
10.30.2.1 _Initialize(GenApi::INodeMap *) . . . . .	562
10.31 CGlobalLock Class Reference . . . . .	562
10.31.1 Detailed Description . . . . .	563
10.31.2 Constructor & Destructor Documentation . . . . .	563
10.31.2.1 CGlobalLock(const char *pszName) . . . . .	563
10.31.2.2 CGlobalLock(const gcstring &strName) . . . . .	563
10.31.2.3 ~CGlobalLock() . . . . .	563
10.31.3 Member Function Documentation . . . . .	563
10.31.3.1 IsValid(void) const . . . . .	563
10.31.3.2 Lock(unsigned int timeout_ms) . . . . .	564
10.31.3.3 TryLock(void) . . . . .	564
10.31.3.4 Unlock(void) . . . . .	564
10.31.4 Member Data Documentation . . . . .	564
10.31.4.1 m_DebugCount . . . . .	564
10.32 CGlobalLockUnlocker Class Reference . . . . .	564
10.32.1 Detailed Description . . . . .	565
10.32.2 Constructor & Destructor Documentation . . . . .	565
10.32.2.1 CGlobalLockUnlocker(CGlobalLock &lock) . . . . .	565
10.32.2.2 ~CGlobalLockUnlocker() . . . . .	565
10.32.3 Member Function Documentation . . . . .	565

10.32.3.1 UnlockEarly(void) . . . . .	565
10.32.4 Member Data Documentation . . . . .	565
10.32.4.1 m_enabled . . . . .	565
10.32.4.2 m_Lock . . . . .	565
10.33 ChunkData Class Reference . . . . .	566
10.33.1 Detailed Description . . . . .	568
10.33.2 Constructor & Destructor Documentation . . . . .	568
10.33.2.1 ChunkData() . . . . .	568
10.33.2.2 ChunkData(const ChunkData &src) . . . . .	568
10.33.2.3 ~ChunkData(void) . . . . .	568
10.33.3 Member Function Documentation . . . . .	568
10.33.3.1 GetBlackLevel() const . . . . .	568
10.33.3.2 GetCounterValue() const . . . . .	568
10.33.3.3 GetCRC() const . . . . .	569
10.33.3.4 GetEncoderValue() const . . . . .	569
10.33.3.5 GetExposureEndLineStatusAll() const . . . . .	569
10.33.3.6 GetExposureTime() const . . . . .	569
10.33.3.7 GetFrameID() const . . . . .	569
10.33.3.8 GetGain() const . . . . .	569
10.33.3.9 GetHeight() const . . . . .	570
10.33.3.10 GetImage() const . . . . .	570
10.33.3.11 GetInferenceConfidence() const . . . . .	570
10.33.3.12 GetInferenceResult() const . . . . .	570
10.33.3.13 GetLinePitch() const . . . . .	570
10.33.3.14 GetLineStatusAll() const . . . . .	570
10.33.3.15 GetOffsetX() const . . . . .	571
10.33.3.16 GetOffsetY() const . . . . .	571
10.33.3.17 GetPartSelector() const . . . . .	571
10.33.3.18 GetPixelDynamicRangeMax() const . . . . .	571
10.33.3.19 GetPixelDynamicRangeMin() const . . . . .	571

10.33.3.20	GetScan3dAxisMax() const	571
10.33.3.21	GetScan3dAxisMin() const	572
10.33.3.22	GetScan3dCoordinateOffset() const	572
10.33.3.23	GetScan3dCoordinateReferenceValue() const	572
10.33.3.24	GetScan3dCoordinateScale() const	572
10.33.3.25	GetScan3dInvalidDataValue() const	572
10.33.3.26	GetScan3dTransformValue() const	572
10.33.3.27	GetScanLineSelector() const	573
10.33.3.28	GetSequencerSetActive() const	573
10.33.3.29	GetSerialDataLength() const	573
10.33.3.30	GetStreamChannelID() const	573
10.33.3.31	GetTimerValue() const	573
10.33.3.32	GetTimestamp() const	573
10.33.3.33	GetTimestampLatchValue() const	574
10.33.3.34	GetTransferBlockID() const	574
10.33.3.35	GetTransferQueueCurrentBlockCount() const	574
10.33.3.36	GetWidth() const	574
10.33.3.37	SetChunks(GenApi::INodeMap &pNodeMap)	574
10.34	CLock Class Reference	575
10.34.1	Detailed Description	575
10.34.2	Constructor & Destructor Documentation	576
10.34.2.1	CLock()	576
10.34.2.2	CLock(void *pLock)	576
10.34.2.3	~CLock()	576
10.34.3	Member Function Documentation	576
10.34.3.1	Lock()	576
10.34.3.2	TryLock()	576
10.34.3.3	Unlock()	576
10.34.4	Friends And Related Function Documentation	576
10.34.4.1	NodeMap	576

10.34.5 Member Data Documentation . . . . .	576
10.34.5.1 m_bOwnLock . . . . .	576
10.34.5.2 m_lock . . . . .	576
10.35 CLock Class Reference . . . . .	577
10.35.1 Detailed Description . . . . .	577
10.35.2 Constructor & Destructor Documentation . . . . .	577
10.35.2.1 CLock() . . . . .	577
10.35.2.2 ~CLock() . . . . .	577
10.35.3 Member Function Documentation . . . . .	578
10.35.3.1 Lock() . . . . .	578
10.35.3.2 TryLock() . . . . .	578
10.35.3.3 Unlock() . . . . .	578
10.36 CLockEx Class Reference . . . . .	578
10.36.1 Detailed Description . . . . .	579
10.36.2 Member Data Documentation . . . . .	579
10.36.2.1 m_lockEx . . . . .	579
10.37 CLockEx Class Reference . . . . .	579
10.37.1 Detailed Description . . . . .	580
10.38 CNodeCallback Class Reference . . . . .	580
10.38.1 Detailed Description . . . . .	581
10.38.2 Constructor & Destructor Documentation . . . . .	581
10.38.2.1 CNodeCallback(INode *pNode, ECallbackType CallbackType) . . . . .	581
10.38.2.2 ~CNodeCallback() . . . . .	581
10.38.3 Member Function Documentation . . . . .	581
10.38.3.1 Destroy()=0 . . . . .	581
10.38.3.2 GetCallbackType() . . . . .	581
10.38.3.3 GetNode() . . . . .	581
10.38.3.4 operator()(ECallbackType CallbackType) const =0 . . . . .	581
10.38.4 Member Data Documentation . . . . .	581
10.38.4.1 m_CallbackType . . . . .	581



10.38.4.2 m_pNode . . . . .	582
10.39 CNodeMapFactory Class Reference . . . . .	582
10.39.1 Detailed Description . . . . .	583
10.39.2 Constructor & Destructor Documentation . . . . .	584
10.39.2.1 CNodeMapFactory() . . . . .	584
10.39.2.2 ~CNodeMapFactory() . . . . .	584
10.39.2.3 CNodeMapFactory(const CNodeMapFactory &) . . . . .	584
10.39.2.4 CNodeMapFactory(EContentType_t FileType, const GenICam::gcstring &File← Name, ECacheUsage_t CacheUsage=CacheUsage_Automatic, bool Suppress← StringsOnLoad=false) . . . . .	584
10.39.2.5 CNodeMapFactory(EContentType_t ContentType, const void *pData, size← _t DataSize, ECacheUsage_t CacheUsage=CacheUsage_Automatic, bool SuppressStringsOnLoad=false) . . . . .	585
10.39.2.6 CNodeMapFactory(const GenICam::gcstring &XmlData, ECacheUsage_← t CacheUsage=CacheUsage_Automatic, bool SuppressStringsOnLoad=false) . . . . .	585
10.39.3 Member Function Documentation . . . . .	586
10.39.3.1 AddInjectionData(CNodeMapFactory &injectionData) . . . . .	586
10.39.3.2 ApplyStyleSheet(const GenICam::gcstring &StyleSheetFileName) . . . . .	586
10.39.3.3 ClearCache() . . . . .	586
10.39.3.4 CreateEmptyNodeMap() . . . . .	586
10.39.3.5 CreateNodeDataFromNodeMap(INodeMap *pNodeMap) . . . . .	587
10.39.3.6 CreateNodeMap(const GenICam::gcstring &DeviceName=""Device"", bool Do← ReleaseCameraDescriptionFileData=true) . . . . .	587
10.39.3.7 CreateNodeMap(CLock &UserProvidedLock, const GenICam::gcstring &← DeviceName=""Device"", bool DoReleaseCameraDescriptionFileData=true) . . . . .	587
10.39.3.8 ExtractSubtree(const GenICam::gcstring &SubTreeRootNodeName, bool do← RenameToRoot=false) . . . . .	587
10.39.3.9 GetNodeStatistics(NodeStatistics_t &NodeStatistics) . . . . .	587
10.39.3.10 GetSupportedSchemaVersions(GenICam::gcstring_vector &SchemaVersions) const . . . . .	587
10.39.3.11 IsCameraDescriptionFileDataReleased() const . . . . .	588
10.39.3.12 IsEmpty() const . . . . .	588
10.39.3.13 IsLoaded() const . . . . .	588
10.39.3.14 IsPreprocessed() const . . . . .	588

10.39.3.15LoadAndInject()	588
10.39.3.16operator=(const CNodeMapFactory &)	588
10.39.3.17Preprocess()	588
10.39.3.18ReleaseCameraDescriptionFileData()	589
10.39.3.19ToString() const	589
10.39.3.20ToXml() const	589
10.40CNodeMapRef Class Reference	589
10.40.1 Detailed Description	590
10.40.2 Constructor & Destructor Documentation	590
10.40.2.1 CNodeMapRef(const GenICam::gcstring &DeviceName=""Device")	590
10.40.2.2 CNodeMapRef(INodeMap *pNodeMap, const GenICam::gcstring &DeviceName=""Device")	591
10.40.2.3 CNodeMapRef(const CNodeMapRef &Them)	591
10.40.3 Member Function Documentation	591
10.40.3.1 operator=(const CNodeMapRef &Them)	591
10.40.3.2 operator=(INodeMap *pNodeMap)	591
10.41CNodeMapRefT< TCameraParams > Class Template Reference	591
10.41.1 Detailed Description	593
10.41.2 Member Function Documentation	593
10.41.2.1 _ClearXMLCache()	593
10.41.2.2 _Connect(IPort *pPort, const GenICam::gcstring &PortName)	593
10.41.2.3 _Connect(IPort *pPort)	594
10.41.2.4 _GetDeviceName()	594
10.41.2.5 _GetNode(const GenICam::gcstring &key)	594
10.41.2.6 _GetNodes(NodeList_t &Nodes)	594
10.41.2.7 _GetSupportedSchemaVersions(GenICam::gcstring_vector &SchemaVersions)	594
10.41.2.8 _InvalidateNodes()	594
10.41.2.9 _LoadXMLFromFile(const GenICam::gcstring &FileName)	594
10.41.2.10 _LoadXMLFromFileInject(const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)	594
10.41.2.11 _LoadXMLFromString(const GenICam::gcstring &XMLData)	594

10.41.2.12_LoadXMLFromStringInject(const GenlCam::gcstring &TargetXMLDataconst, const GenlCam::gcstring &InjectXMLData) . . . . .	595
10.41.2.13_LoadXMLFromZIPData(const void *zipData, size_t zipSize) . . . . .	595
10.41.2.14_LoadXMLFromZIPFile(const GenlCam::gcstring &ZipFileName) . . . . .	595
10.41.2.15_Poll(int64_t ElapsedTime) . . . . .	595
10.41.3 Member Data Documentation . . . . .	595
10.41.3.1 _Ptr . . . . .	595
10.42CommandNode Class Reference . . . . .	595
10.42.1 Detailed Description . . . . .	596
10.42.2 Constructor & Destructor Documentation . . . . .	597
10.42.2.1 CommandNode() . . . . .	597
10.42.2.2 CommandNode(std::shared_ptr< Node::NodeImpl > pCommand) . . . . .	597
10.42.2.3 ~CommandNode() . . . . .	597
10.42.3 Member Function Documentation . . . . .	597
10.42.3.1 Execute(bool Verify=true) . . . . .	597
10.42.3.2 IsDone(bool Verify=true) . . . . .	597
10.42.3.3 operator>() . . . . .	597
10.42.3.4 SetReference(INode *pBase) . . . . .	597
10.43Counter Class Reference . . . . .	598
10.43.1 Detailed Description . . . . .	598
10.43.2 Constructor & Destructor Documentation . . . . .	598
10.43.2.1 Counter() . . . . .	598
10.43.3 Member Function Documentation . . . . .	598
10.43.3.1 GetValue() const . . . . .	598
10.43.3.2 IsZero() . . . . .	598
10.43.3.3 operator unsigned int() . . . . .	598
10.43.3.4 operator++() . . . . .	598
10.43.3.5 operator++(int) . . . . .	598
10.43.3.6 operator--(int) . . . . .	598
10.43.3.7 operator--() . . . . .	598
10.44CPointer< T, B > Class Template Reference . . . . .	599

10.44.1 Detailed Description . . . . .	600
10.44.2 Constructor & Destructor Documentation . . . . .	600
10.44.2.1 CPointer(void) . . . . .	600
10.44.2.2 CPointer(B *pB) . . . . .	600
10.44.2.3 ~CPointer(void) . . . . .	600
10.44.3 Member Function Documentation . . . . .	600
10.44.3.1 IsValid() const . . . . .	600
10.44.3.2 operator bool(void) const . . . . .	600
10.44.3.3 operator T *(void) const . . . . .	600
10.44.3.4 operator!=(const CPointer< T, B > &rT) const . . . . .	601
10.44.3.5 operator!=(T *pT) const . . . . .	601
10.44.3.6 operator!=(const long int nMustBeNull) const . . . . .	601
10.44.3.7 operator!=(const int nMustBeNull) const . . . . .	601
10.44.3.8 operator()(void) const . . . . .	601
10.44.3.9 operator*(void) const . . . . .	601
10.44.3.10 operator->(void) const . . . . .	601
10.44.3.11 operator=(B *pB) . . . . .	601
10.44.3.12 operator==(T *pT) const . . . . .	601
10.44.3.13 operator==(const CPointer< T, B > &rT) const . . . . .	601
10.44.3.14 operator==(int nMustBeNull) const . . . . .	602
10.44.4 Member Data Documentation . . . . .	602
10.44.4.1 m_pT . . . . .	602
10.45 CPortImpl Class Reference . . . . .	602
10.45.1 Detailed Description . . . . .	603
10.45.2 Constructor & Destructor Documentation . . . . .	603
10.45.2.1 CPortImpl() . . . . .	603
10.45.2.2 ~CPortImpl() . . . . .	603
10.45.3 Member Function Documentation . . . . .	603
10.45.3.1 GetAccessMode() const =0 . . . . .	603
10.45.3.2 GetSwapEndianness() . . . . .	604

10.45.3.3 InvalidateNode()	604
10.45.3.4 Read(void *pBuffer, int64_t Address, int64_t Length)=0	604
10.45.3.5 Replay(IPortWriteList *pPortRecorder, bool Invalidate=true)	604
10.45.3.6 SetPortImpl(IPort *pPort)	604
10.45.3.7 Write(const void *pBuffer, int64_t Address, int64_t Length)=0	604
10.45.4 Member Data Documentation	604
10.45.4.1 m_ptrPort	604
10.46 CPortWriteList Class Reference	605
10.46.1 Detailed Description	606
10.46.2 Constructor & Destructor Documentation	606
10.46.2.1 CPortWriteList()	606
10.46.2.2 ~CPortWriteList()	606
10.46.3 Member Function Documentation	606
10.46.3.1 GetCookie()	606
10.46.3.2 GetPortWriteListHandle()	606
10.46.3.3 Replay(IPort *pPort)	606
10.46.3.4 SetCookie(const int64_t Value)	606
10.46.3.5 Write(const void *pBuffer, int64_t Address, int64_t Length)	606
10.46.4 Member Data Documentation	607
10.46.4.1 m_pWriteList	607
10.47 CRegisterPortImpl Class Reference	607
10.47.1 Detailed Description	608
10.47.2 Constructor & Destructor Documentation	608
10.47.2.1 CRegisterPortImpl(int MaxNumQuadlets=1, bool TransportLayerSwaps← Endianness=false)	608
10.47.2.2 ~CRegisterPortImpl()	608
10.47.3 Member Function Documentation	608
10.47.3.1 GetAccessMode() const =0	608
10.47.3.2 Read(void *pBuffer, int64_t Address, int64_t Length)	609
10.47.3.3 ReadRegister(uint32_t *pRegisters, int64_t Address, int64_t Length)=0	609
10.47.3.4 SetPortImpl(IPort *pPort)	609

10.47.3.5 Write(const void *pBuffer, int64_t Address, int64_t Length) . . . . .	609
10.47.3.6 WriteRegister(const uint32_t *pRegisters, int64_t Address, int64_t Length)=0 . . . . .	609
10.48 CSelectorSet Class Reference . . . . .	609
10.48.1 Detailed Description . . . . .	610
10.48.2 Constructor & Destructor Documentation . . . . .	610
10.48.2.1 CSelectorSet(IBase *pBase) . . . . .	610
10.48.2.2 ~CSelectorSet() . . . . .	611
10.48.3 Member Function Documentation . . . . .	611
10.48.3.1 GetSelectorList(FeatureList_t &SelectorList, bool Incremental=false) . . . . .	611
10.48.3.2 IsEmpty() . . . . .	611
10.48.3.3 Restore() . . . . .	611
10.48.3.4 SetFirst() . . . . .	611
10.48.3.5 SetNext(bool Tick=true) . . . . .	611
10.48.3.6 ToString() . . . . .	611
10.49 CTestPortStruct< CDataStruct > Class Template Reference . . . . .	611
10.49.1 Detailed Description . . . . .	613
10.49.2 Constructor & Destructor Documentation . . . . .	613
10.49.2.1 CTestPortStruct(int64_t BaseAddress=0) . . . . .	613
10.49.3 Member Function Documentation . . . . .	613
10.49.3.1 GetAccessMode() const . . . . .	613
10.49.3.2 GetNumReads() . . . . .	613
10.49.3.3 GetNumWrites() . . . . .	613
10.49.3.4 GetPrincipalInterfaceType() const . . . . .	614
10.49.3.5 MemSet(const char FillValue) . . . . .	614
10.49.3.6 Read(void *pBuffer, int64_t Address, int64_t Length) . . . . .	614
10.49.3.7 ResetStatistics() . . . . .	614
10.49.3.8 Write(const void *pBuffer, int64_t Address, int64_t Length) . . . . .	614
10.49.4 Member Data Documentation . . . . .	614
10.49.4.1 m_BaseAddress . . . . .	614
10.49.4.2 m_NumReads . . . . .	614

10.49.4.3 m_NumWrites . . . . .	614
10.50DCAM_CHECKSUM Struct Reference . . . . .	614
10.50.1 Member Data Documentation . . . . .	615
10.50.1.1 CRCChecksum . . . . .	615
10.51DCAM_CHUNK_TRAILER Struct Reference . . . . .	615
10.51.1 Member Data Documentation . . . . .	615
10.51.1.1 ChunkID . . . . .	615
10.51.1.2 ChunkLength . . . . .	615
10.51.1.3 InverseChunkLength . . . . .	615
10.52DeviceEvent Class Reference . . . . .	615
10.52.1 Detailed Description . . . . .	616
10.52.2 Constructor & Destructor Documentation . . . . .	616
10.52.2.1 DeviceEvent() . . . . .	616
10.52.2.2 ~DeviceEvent() . . . . .	617
10.52.3 Member Function Documentation . . . . .	617
10.52.3.1 GetDeviceEventId() const . . . . .	617
10.52.3.2 GetDeviceEventName() const . . . . .	617
10.52.3.3 OnDeviceEvent(Spinnaker::GenICam::gcstring eventName)=0 . . . . .	617
10.52.3.4 operator=(const DeviceEvent &) . . . . .	617
10.53double_autovector_t Class Reference . . . . .	618
10.53.1 Detailed Description . . . . .	618
10.53.2 Constructor & Destructor Documentation . . . . .	618
10.53.2.1 double_autovector_t() . . . . .	618
10.53.2.2 double_autovector_t(const double_autovector_t &obj) . . . . .	618
10.53.2.3 double_autovector_t(size_t n) . . . . .	618
10.53.2.4 ~double_autovector_t(void) . . . . .	618
10.53.3 Member Function Documentation . . . . .	618
10.53.3.1 operator delete(void *pWhere) . . . . .	618
10.53.3.2 operator new(size_t uiSize) . . . . .	618
10.53.3.3 operator=(const double_autovector_t &obj) . . . . .	618

10.53.3.4 operator[](size_t uiIndex) . . . . .	618
10.53.3.5 operator[](size_t uiIndex) const . . . . .	619
10.53.3.6 size() const . . . . .	619
10.53.4 Member Data Documentation . . . . .	619
10.53.4.1 _pCount . . . . .	619
10.53.4.2 _pv . . . . .	619
10.54EAccessModeClass Class Reference . . . . .	619
10.54.1 Detailed Description . . . . .	619
10.54.2 Member Function Documentation . . . . .	619
10.54.2.1 FromString(const GenICam::gcstring &ValueStr, EAccessMode *pValue) . . . . .	619
10.54.2.2 ToString(GenICam::gcstring &ValueStr, EAccessMode *pValue) . . . . .	619
10.54.2.3 ToString(EAccessMode Value) . . . . .	620
10.55ECachingModeClass Class Reference . . . . .	620
10.55.1 Detailed Description . . . . .	620
10.55.2 Member Function Documentation . . . . .	620
10.55.2.1 FromString(const GenICam::gcstring &ValueStr, ECachingMode *pValue) . . . . .	620
10.55.2.2 ToString(GenICam::gcstring &ValueStr, ECachingMode *pValue) . . . . .	620
10.55.2.3 ToString(ECachingMode Value) . . . . .	620
10.56EDisplayNotationClass Class Reference . . . . .	620
10.56.1 Detailed Description . . . . .	621
10.56.2 Member Function Documentation . . . . .	621
10.56.2.1 FromString(const GenICam::gcstring &ValueStr, EDisplayNotation *pValue) . . . . .	621
10.56.2.2 ToString(GenICam::gcstring &ValueStr, EDisplayNotation *pValue) . . . . .	621
10.56.2.3 ToString(EDisplayNotation Value) . . . . .	621
10.57EEndianessClass Class Reference . . . . .	621
10.57.1 Detailed Description . . . . .	622
10.57.2 Member Function Documentation . . . . .	622
10.57.2.1 FromString(const GenICam::gcstring &ValueStr, EEndianess *pValue) . . . . .	622
10.57.2.2 ToString(GenICam::gcstring &ValueStr, EEndianess *pValue) . . . . .	622
10.57.2.3 ToString(EEndianess Value) . . . . .	622



10.58 EGenApiSchemaVersionClass Class Reference . . . . .	622
10.58.1 Detailed Description . . . . .	622
10.58.2 Member Function Documentation . . . . .	623
10.58.2.1 FromString(const GenICam::gcstring &ValueStr, EGenApiSchemaVersion *pValue)	623
10.58.2.2 ToString(GenICam::gcstring &ValueStr, EGenApiSchemaVersion *pValue) . . . .	623
10.58.2.3 ToString(EGenApiSchemaVersion Value) . . . . .	623
10.59 EInputDirectionClass Class Reference . . . . .	623
10.59.1 Detailed Description . . . . .	623
10.59.2 Member Function Documentation . . . . .	623
10.59.2.1 FromString(const GenICam::gcstring &ValueStr, EInputDirection *pValue) . . . .	623
10.59.2.2 ToString(GenICam::gcstring &ValueStr, EInputDirection *pValue) . . . . .	624
10.59.2.3 ToString(EInputDirection Value) . . . . .	624
10.60 ENameSpaceClass Class Reference . . . . .	624
10.60.1 Detailed Description . . . . .	624
10.60.2 Member Function Documentation . . . . .	624
10.60.2.1 FromString(const GenICam::gcstring &ValueStr, ENameSpace *pValue) . . . .	624
10.60.2.2 ToString(GenICam::gcstring &ValueStr, ENameSpace *pValue) . . . . .	624
10.60.2.3 ToString(ENameSpace Value) . . . . .	625
10.61 EnumEntryNode Class Reference . . . . .	625
10.61.1 Detailed Description . . . . .	626
10.61.2 Constructor & Destructor Documentation . . . . .	627
10.61.2.1 EnumEntryNode() . . . . .	627
10.61.2.2 EnumEntryNode(std::shared_ptr< Node::NodeImpl > pEnumEntry) . . . . .	627
10.61.2.3 ~EnumEntryNode() . . . . .	627
10.61.3 Member Function Documentation . . . . .	627
10.61.3.1 GetNumericValue() . . . . .	627
10.61.3.2 GetSymbolic() const . . . . .	627
10.61.3.3 GetValue() . . . . .	627
10.61.3.4 IsSelfClearing() . . . . .	627
10.61.3.5 SetReference(INode *pBase) . . . . .	627

10.62EnumNode Class Reference . . . . .	628
10.62.1 Detailed Description . . . . .	630
10.62.2 Constructor & Destructor Documentation . . . . .	630
10.62.2.1 EnumNode() . . . . .	630
10.62.2.2 EnumNode(std::shared_ptr< Node::NodeImpl > pEnumeration) . . . . .	630
10.62.2.3 ~EnumNode() . . . . .	630
10.62.3 Member Function Documentation . . . . .	630
10.62.3.1 GetCurrentEntry(bool Verify=false, bool IgnoreCache=false) . . . . .	630
10.62.3.2 GetEntries(NodeList_t &Entries) . . . . .	630
10.62.3.3 GetEntry(const int64_t IntValue) . . . . .	630
10.62.3.4 GetEntryByName(const GenICam::gcstring &Symbolic) . . . . .	630
10.62.3.5 GetIntValue(bool Verify=false, bool IgnoreCache=false) . . . . .	630
10.62.3.6 GetSymbolics(StringList_t &Symbolics) . . . . .	631
10.62.3.7 operator*() . . . . .	631
10.62.3.8 operator=(const GenICam::gcstring &ValueStr) . . . . .	631
10.62.3.9 SetIntValue(int64_t Value, bool Verify=true) . . . . .	631
10.62.3.10SetReference(INode *pBase) . . . . .	631
10.62.4 Member Data Documentation . . . . .	631
10.62.4.1 m_pEnumeration . . . . .	631
10.63ERepresentationClass Class Reference . . . . .	632
10.63.1 Detailed Description . . . . .	632
10.63.2 Member Function Documentation . . . . .	632
10.63.2.1 FromString(const GenICam::gcstring &ValueStr, ERepresentation *pValue) . . . . .	632
10.63.2.2 ToString(GenICam::gcstring &ValueStr, ERepresentation *pValue) . . . . .	632
10.63.2.3 ToString(ERepresentation Value) . . . . .	632
10.64ESignClass Class Reference . . . . .	632
10.64.1 Detailed Description . . . . .	633
10.64.2 Member Function Documentation . . . . .	633
10.64.2.1 FromString(const GenICam::gcstring &ValueStr, ESign *pValue) . . . . .	633
10.64.2.2 ToString(GenICam::gcstring &ValueStr, ESign *pValue) . . . . .	633

10.64.2.3 ToString(ESign Value) . . . . .	633
10.65ESlopeClass Class Reference . . . . .	633
10.65.1 Detailed Description . . . . .	634
10.65.2 Member Function Documentation . . . . .	634
10.65.2.1 FromString(const GenICam::gcstring &ValueStr, ESlope *pValue) . . . . .	634
10.65.2.2 ToString(GenICam::gcstring &ValueStr, ESlope *pValue) . . . . .	634
10.65.2.3 ToString(ESlope Value) . . . . .	634
10.66EStandardNameSpaceClass Class Reference . . . . .	634
10.66.1 Detailed Description . . . . .	634
10.66.2 Member Function Documentation . . . . .	635
10.66.2.1 FromString(const GenICam::gcstring &ValueStr, EStandardNameSpace *pValue) . . . . .	635
10.66.2.2 ToString(GenICam::gcstring &ValueStr, EStandardNameSpace *pValue) . . . . .	635
10.66.2.3 ToString(EStandardNameSpace Value) . . . . .	635
10.67Event Class Reference . . . . .	635
10.67.1 Detailed Description . . . . .	636
10.67.2 Constructor & Destructor Documentation . . . . .	636
10.67.2.1 ~Event() . . . . .	636
10.67.2.2 Event() . . . . .	636
10.67.3 Member Function Documentation . . . . .	636
10.67.3.1 GetEventPayloadData() . . . . .	636
10.67.3.2 GetEventPayloadDataSize() . . . . .	636
10.67.3.3 GetEventType() . . . . .	637
10.67.3.4 operator=(const Event &) . . . . .	637
10.67.3.5 SetEventPayload(uint8_t *offset, size_t length) . . . . .	637
10.67.3.6 SetEventType(EventType eventType) . . . . .	637
10.67.4 Friends And Related Function Documentation . . . . .	637
10.67.4.1 EventProcessor . . . . .	637
10.67.4.2 IDataStream . . . . .	637
10.67.4.3 Stream . . . . .	637
10.67.5 Member Data Documentation . . . . .	637

10.67.5.1 m_pEventData . . . . .	637
10.68EVisibilityClass Class Reference . . . . .	637
10.68.1 Detailed Description . . . . .	638
10.68.2 Member Function Documentation . . . . .	638
10.68.2.1 FromString(const GenICam::gcstring &ValueStr, EVisibility *pValue) . . . . .	638
10.68.2.2 ToString(GenICam::gcstring &ValueStr, EVisibility *pValue) . . . . .	638
10.68.2.3 ToString(EVisibility Value) . . . . .	638
10.69Exception Class Reference . . . . .	638
10.69.1 Detailed Description . . . . .	639
10.69.2 Constructor & Destructor Documentation . . . . .	640
10.69.2.1 Exception() . . . . .	640
10.69.2.2 Exception(int line, const char *fileName, const char *funcName, const char *buildDate, const char *buildTime, const char *errMsg, Error err) . . . . .	640
10.69.2.3 Exception(const Exception &except) . . . . .	640
10.69.2.4 ~Exception() . . . . .	640
10.69.3 Member Function Documentation . . . . .	640
10.69.3.1 GetBuildDate() const . . . . .	640
10.69.3.2 GetBuildTime() const . . . . .	640
10.69.3.3 GetError() const . . . . .	640
10.69.3.4 GetErrorMessage() const . . . . .	640
10.69.3.5 GetFileName() const . . . . .	640
10.69.3.6 GetFullErrorMessage() const . . . . .	640
10.69.3.7 GetFunctionName() const . . . . .	641
10.69.3.8 GetLineNumber() const . . . . .	641
10.69.3.9 operator!=(const Error err) const . . . . .	641
10.69.3.10operator=(const Exception &except) . . . . .	641
10.69.3.11operator==(const Error err) const . . . . .	641
10.69.3.12what() const . . . . .	641
10.70EYesNoClass Class Reference . . . . .	641
10.70.1 Detailed Description . . . . .	641
10.70.2 Member Function Documentation . . . . .	642

10.70.2.1 FromString(const GenICam::gcstring &ValueStr, EYesNo *pValue) . . . . .	642
10.70.2.2 ToString(GenICam::gcstring &ValueStr, EYesNo *pValue) . . . . .	642
10.70.2.3 ToString(EYesNo Value) . . . . .	642
10.71 FileProtocolAdapter Class Reference . . . . .	642
10.71.1 Detailed Description . . . . .	642
10.71.2 Constructor & Destructor Documentation . . . . .	643
10.71.2.1 FileProtocolAdapter() . . . . .	643
10.71.2.2 ~FileProtocolAdapter() . . . . .	643
10.71.3 Member Function Documentation . . . . .	643
10.71.3.1 attach(::Spinnaker::GenApi::INodeMap *pInterface) . . . . .	643
10.71.3.2 closeFile(const char *pFileName) . . . . .	643
10.71.3.3 deleteFile(const char *pFileName) . . . . .	643
10.71.3.4 getBufSize(const char *pFileName, std::ios_base::openmode mode) . . . . .	644
10.71.3.5 openFile(const char *pFileName, std::ios_base::openmode mode) . . . . .	644
10.71.3.6 read(char *buf, int64_t offs, std::streamsize len, const char *pFileName) . . . . .	644
10.71.3.7 write(const char *buf, int64_t offs, int64_t len, const char *pFileName) . . . . .	645
10.72 FloatNode Class Reference . . . . .	645
10.72.1 Detailed Description . . . . .	647
10.72.2 Constructor & Destructor Documentation . . . . .	647
10.72.2.1 FloatNode() . . . . .	647
10.72.2.2 FloatNode(std::shared_ptr< Node::NodeImpl > pFloat) . . . . .	647
10.72.2.3 ~FloatNode() . . . . .	647
10.72.3 Member Function Documentation . . . . .	647
10.72.3.1 GetDisplayNotation() const . . . . .	647
10.72.3.2 GetDisplayPrecision() const . . . . .	647
10.72.3.3 GetEnumAlias() . . . . .	647
10.72.3.4 GetInc() . . . . .	648
10.72.3.5 GetIncMode() . . . . .	648
10.72.3.6 GetIntAlias() . . . . .	648
10.72.3.7 GetListOfValidValues(bool bounded=true) . . . . .	648

10.72.3.8 GetMax()	648
10.72.3.9 GetMin()	648
10.72.3.10 GetRepresentation()	648
10.72.3.11 GetUnit() const	648
10.72.3.12 GetValue(bool Verify=false, bool IgnoreCache=false)	648
10.72.3.13 HasInc()	649
10.72.3.14 ImposeMax(double Value)	649
10.72.3.15 ImposeMin(double Value)	649
10.72.3.16 operator()()	649
10.72.3.17 operator*()	649
10.72.3.18 operator=(double Value)	649
10.72.3.19 SetReference(INode *pBase)	649
10.72.3.20 SetValue(double Value, bool Verify=true)	649
10.73 FloatRegNode Class Reference	650
10.73.1 Detailed Description	651
10.73.2 Constructor & Destructor Documentation	651
10.73.2.1 FloatRegNode()	651
10.73.2.2 FloatRegNode(std::shared_ptr< Node::NodeImpl > pFloat)	651
10.73.2.3 ~FloatRegNode()	652
10.73.3 Member Function Documentation	652
10.73.3.1 SetReference(INode *pBase)	652
10.74 Function_NodeCallback< Function > Class Template Reference	652
10.74.1 Detailed Description	653
10.74.2 Constructor & Destructor Documentation	653
10.74.2.1 Function_NodeCallback(INode *pNode, const Function &function, ECallbackType CallbackType)	653
10.74.3 Member Function Documentation	653
10.74.3.1 Destroy()	653
10.74.3.2 operator()(ECallbackType CallbackType) const	653
10.75 gcstring Class Reference	654
10.75.1 Constructor & Destructor Documentation	655

10.75.1.1	<code>gcstring()</code>	655
10.75.1.2	<code>gcstring(const char *pc)</code>	655
10.75.1.3	<code>gcstring(const char *pc, size_t n)</code>	655
10.75.1.4	<code>gcstring(size_t count, char ch)</code>	655
10.75.1.5	<code>gcstring(const gcstring &amp;str)</code>	655
10.75.1.6	<code>~gcstring(void)</code>	655
10.75.2	Member Function Documentation	655
10.75.2.1	<code>_npos(void)</code>	655
10.75.2.2	<code>append(const gcstring &amp;str)</code>	655
10.75.2.3	<code>append(size_t count, char ch)</code>	655
10.75.2.4	<code>assign(const gcstring &amp;str)</code>	655
10.75.2.5	<code>assign(size_t count, char ch)</code>	655
10.75.2.6	<code>assign(const char *pc)</code>	655
10.75.2.7	<code>assign(const char *pc, size_t n)</code>	655
10.75.2.8	<code>c_str(void) const</code>	655
10.75.2.9	<code>compare(const gcstring &amp;str) const</code>	655
10.75.2.10	<code>empty(void) const</code>	655
10.75.2.11	<code>find(char ch, size_t offset=0) const</code>	656
10.75.2.12	<code>find(const gcstring &amp;str, size_t offset=0) const</code>	656
10.75.2.13	<code>find(const gcstring &amp;str, size_t offset, size_t count) const</code>	656
10.75.2.14	<code>find(const char *pc, size_t offset=0) const</code>	656
10.75.2.15	<code>find(const char *pc, size_t offset, size_t count) const</code>	656
10.75.2.16	<code>find_first_not_of(const gcstring &amp;str, size_t offset=0) const</code>	656
10.75.2.17	<code>find_first_of(const gcstring &amp;str, size_t offset=0) const</code>	656
10.75.2.18	<code>length(void) const</code>	656
10.75.2.19	<code>max_size() const</code>	656
10.75.2.20	<code>operator const char *(void) const</code>	656
10.75.2.21	<code>operator delete(void *pWhere)</code>	656
10.75.2.22	<code>operator delete(void *pWhere, void *pNewWhere)</code>	656
10.75.2.23	<code>operator new(size_t uiSize)</code>	656

10.75.2.24	<code>operator new(size_t uiSize, void *pWhere)</code>	656
10.75.2.25	<code>operator!=(const gcstring &amp;str) const</code>	656
10.75.2.26	<code>operator!=(const char *pc) const</code>	656
10.75.2.27	<code>operator+=(const gcstring &amp;str)</code>	656
10.75.2.28	<code>operator+=(const gcstring &amp;str) const</code>	656
10.75.2.29	<code>operator+=(const char *pc)</code>	656
10.75.2.30	<code>operator+=(char ch)</code>	656
10.75.2.31	<code>operator+=(char ch) const</code>	656
10.75.2.32	<code>operator&lt;(const gcstring &amp;str) const</code>	656
10.75.2.33	<code>operator=(const gcstring &amp;str)</code>	656
10.75.2.34	<code>operator==(const gcstring &amp;str) const</code>	657
10.75.2.35	<code>operator==(const char *pc) const</code>	657
10.75.2.36	<code>operator&gt;(const gcstring &amp;str) const</code>	657
10.75.2.37	<code>resize(size_t n)</code>	657
10.75.2.38	<code>size(void) const</code>	657
10.75.2.39	<code>substr(size_t offset=0, size_t count=GCSTRING_NPOS) const</code>	657
10.75.2.40	<code>swap(gcstring &amp;Right)</code>	657
10.75.3	Friends And Related Function Documentation	657
10.75.3.1	<code>operator+</code>	657
10.75.3.2	<code>operator+</code>	657
10.75.3.3	<code>operator+</code>	657
10.75.4	Member Data Documentation	657
10.75.4.1	<code>npos</code>	657
10.76	GVCP_CHUNK_TRAILER Struct Reference	657
10.76.1	Detailed Description	657
10.76.2	Member Data Documentation	658
10.76.2.1	<code>ChunkID</code>	658
10.76.2.2	<code>ChunkLength</code>	658
10.77	GVCP_EVENT_ITEM Struct Reference	658
10.77.1	Detailed Description	658



10.77.2 Member Data Documentation . . . . .	658
10.77.2.1 BlockId . . . . .	658
10.77.2.2 EventId . . . . .	658
10.77.2.3 ReservedOrEventSize . . . . .	658
10.77.2.4 StreamChannelId . . . . .	658
10.77.2.5 TimestampHigh . . . . .	658
10.77.2.6 TimestampLow . . . . .	658
10.78GVCP_EVENT_ITEM_BASIC Struct Reference . . . . .	659
10.78.1 Detailed Description . . . . .	659
10.78.2 Member Data Documentation . . . . .	659
10.78.2.1 EventId . . . . .	659
10.78.2.2 ReservedOrEventSize . . . . .	659
10.79GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference . . . . .	659
10.79.1 Detailed Description . . . . .	659
10.79.2 Member Data Documentation . . . . .	660
10.79.2.1 BlockId . . . . .	660
10.79.2.2 BlockId64High . . . . .	660
10.79.2.3 BlockId64Low . . . . .	660
10.79.2.4 EventId . . . . .	660
10.79.2.5 ReservedOrEventSize . . . . .	660
10.79.2.6 StreamChannelId . . . . .	660
10.79.2.7 TimestampHigh . . . . .	660
10.79.2.8 TimestampLow . . . . .	660
10.80GVCP_EVENT_REQUEST Struct Reference . . . . .	660
10.80.1 Detailed Description . . . . .	661
10.80.2 Member Data Documentation . . . . .	661
10.80.2.1 Header . . . . .	661
10.80.2.2 Items . . . . .	661
10.81GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference . . . . .	661
10.81.1 Detailed Description . . . . .	661

10.81.2 Member Data Documentation . . . . .	662
10.81.2.1 Header . . . . .	662
10.81.2.2 Items . . . . .	662
10.82GVCP_EVENTDATA_REQUEST Struct Reference . . . . .	662
10.82.1 Detailed Description . . . . .	662
10.82.2 Member Data Documentation . . . . .	662
10.82.2.1 Data . . . . .	662
10.82.2.2 Event . . . . .	662
10.82.2.3 Header . . . . .	662
10.83GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference . . . . .	663
10.83.1 Detailed Description . . . . .	663
10.83.2 Member Data Documentation . . . . .	663
10.83.2.1 Data . . . . .	663
10.83.2.2 Event . . . . .	663
10.83.2.3 Header . . . . .	663
10.84GVCP_REQUEST_HEADER Struct Reference . . . . .	663
10.84.1 Detailed Description . . . . .	664
10.84.2 Member Data Documentation . . . . .	664
10.84.2.1 Command . . . . .	664
10.84.2.2 Flags . . . . .	664
10.84.2.3 Length . . . . .	664
10.84.2.4 Magic . . . . .	664
10.84.2.5 ReqId . . . . .	664
10.85H264Option Struct Reference . . . . .	664
10.85.1 Detailed Description . . . . .	665
10.85.2 Constructor & Destructor Documentation . . . . .	665
10.85.2.1 H264Option() . . . . .	665
10.85.3 Member Data Documentation . . . . .	665
10.85.3.1 bitrate . . . . .	665
10.85.3.2 frameRate . . . . .	665

10.85.3.3 height	665
10.85.3.4 reserved	665
10.85.3.5 width	665
10.86 IArrivalEvent Class Reference	666
10.86.1 Constructor & Destructor Documentation	667
10.86.1.1 ~IArrivalEvent()	667
10.86.1.2 IArrivalEvent()	667
10.86.1.3 IArrivalEvent(const IArrivalEvent &)	667
10.86.2 Member Function Documentation	667
10.86.2.1 OnDeviceArrival(uint64_t serialNumber)=0	667
10.86.2.2 operator=(const IArrivalEvent &)	667
10.87 ICameraBase Class Reference	667
10.87.1 Detailed Description	669
10.87.2 Constructor & Destructor Documentation	669
10.87.2.1 ~ICameraBase(void)	669
10.87.2.2 ICameraBase()	669
10.87.2.3 ICameraBase(const ICameraBase &)	669
10.87.3 Member Function Documentation	669
10.87.3.1 BeginAcquisition()=0	669
10.87.3.2 DeInit()=0	669
10.87.3.3 DiscoverMaxPacketSize()=0	669
10.87.3.4 EndAcquisition()=0	669
10.87.3.5 GetAccessMode() const =0	670
10.87.3.6 GetGuiXml() const =0	670
10.87.3.7 GetNextImage(uint64_t grabTimeout=EVENT_TIMEOUT_INFINITE, uint64_t streamID=0)=0	670
10.87.3.8 GetNodeMap() const =0	670
10.87.3.9 GetNumDataStreams()=0	670
10.87.3.10 GetNumImagesInUse()=0	670
10.87.3.11 GetTLDeviceNodeMap() const =0	670
10.87.3.12 GetTLStreamNodeMap() const =0	670

10.87.3.13	GetUniqueID()=0 . . . . .	670
10.87.3.14	Init()=0 . . . . .	670
10.87.3.15	IsInitialized()=0 . . . . .	671
10.87.3.16	IsStreaming() const =0 . . . . .	671
10.87.3.17	IsValid()=0 . . . . .	671
10.87.3.18	operator=(const ICameraBase &) . . . . .	671
10.87.3.19	ReadPort(uint64_t iAddress, void *pBuffer, size_t iSize)=0 . . . . .	671
10.87.3.20	RegisterEvent(Event &evtToRegister)=0 . . . . .	671
10.87.3.21	RegisterEvent(Event &evtToRegister, const GenICam::gcstring &eventName)=0 . . . . .	671
10.87.3.22	UnregisterEvent(Event &evtToUnregister)=0 . . . . .	671
10.87.3.23	WritePort(uint64_t iAddress, const void *pBuffer, size_t iSize)=0 . . . . .	671
10.87.4	Friends And Related Function Documentation . . . . .	672
10.87.4.1	CameraInternal . . . . .	672
10.87.4.2	InterfaceImpl . . . . .	672
10.87.5	Member Data Documentation . . . . .	672
10.87.5.1	m_pCameraBaseData . . . . .	672
10.87.5.2	TLDevice . . . . .	672
10.87.5.3	TLStream . . . . .	672
10.88	ICameraList Class Reference . . . . .	672
10.88.1	Detailed Description . . . . .	673
10.88.2	Constructor & Destructor Documentation . . . . .	673
10.88.2.1	~ICameraList() . . . . .	673
10.88.2.2	ICameraList() . . . . .	673
10.88.2.3	ICameraList(const ICameraList &) . . . . .	673
10.88.3	Member Function Documentation . . . . .	673
10.88.3.1	Append(CameraList &otherList)=0 . . . . .	673
10.88.3.2	Clear()=0 . . . . .	674
10.88.3.3	GetByIndex(unsigned int index) const =0 . . . . .	674
10.88.3.4	GetBySerial(std::string serialNumber) const =0 . . . . .	674
10.88.3.5	GetSize() const =0 . . . . .	674

10.88.3.6 operator=(const ICameraList &) . . . . .	674
10.88.3.7 operator[](unsigned int index)=0 . . . . .	674
10.88.3.8 RemoveByIndex(unsigned int index)=0 . . . . .	674
10.88.3.9 RemoveBySerial(std::string serialNumber)=0 . . . . .	674
10.88.4 Friends And Related Function Documentation . . . . .	674
10.88.4.1 CameraListImpl . . . . .	674
10.88.4.2 InterfaceImpl . . . . .	674
10.88.5 Member Data Documentation . . . . .	674
10.88.5.1 m_pCameraListData . . . . .	674
10.89 IChunkData Class Reference . . . . .	675
10.89.1 Detailed Description . . . . .	676
10.89.2 Constructor & Destructor Documentation . . . . .	676
10.89.2.1 ~IChunkData() . . . . .	676
10.89.2.2 IChunkData() . . . . .	676
10.89.3 Member Function Documentation . . . . .	676
10.89.3.1 GetBlackLevel() const =0 . . . . .	676
10.89.3.2 GetCounterValue() const =0 . . . . .	676
10.89.3.3 GetCRC() const =0 . . . . .	676
10.89.3.4 GetEncoderValue() const =0 . . . . .	676
10.89.3.5 GetExposureEndLineStatusAll() const =0 . . . . .	676
10.89.3.6 GetExposureTime() const =0 . . . . .	677
10.89.3.7 GetFrameID() const =0 . . . . .	677
10.89.3.8 GetGain() const =0 . . . . .	677
10.89.3.9 GetHeight() const =0 . . . . .	677
10.89.3.10 GetImage() const =0 . . . . .	677
10.89.3.11 GetInferenceConfidence() const =0 . . . . .	677
10.89.3.12 GetInferenceResult() const =0 . . . . .	677
10.89.3.13 GetLinePitch() const =0 . . . . .	677
10.89.3.14 GetLineStatusAll() const =0 . . . . .	677
10.89.3.15 GetOffsetX() const =0 . . . . .	677

10.89.3.16	GetOffsetY() const =0	678
10.89.3.17	GetPartSelector() const =0	678
10.89.3.18	GetPixelDynamicRangeMax() const =0	678
10.89.3.19	GetPixelDynamicRangeMin() const =0	678
10.89.3.20	GetScan3dAxisMax() const =0	678
10.89.3.21	GetScan3dAxisMin() const =0	678
10.89.3.22	GetScan3dCoordinateOffset() const =0	678
10.89.3.23	GetScan3dCoordinateReferenceValue() const =0	678
10.89.3.24	GetScan3dCoordinateScale() const =0	678
10.89.3.25	GetScan3dInvalidDataValue() const =0	678
10.89.3.26	GetScan3dTransformValue() const =0	679
10.89.3.27	GetScanLineSelector() const =0	679
10.89.3.28	GetSequencerSetActive() const =0	679
10.89.3.29	GetSerialDataLength() const =0	679
10.89.3.30	GetStreamChannelID() const =0	679
10.89.3.31	GetTimerValue() const =0	679
10.89.3.32	GetTimestamp() const =0	679
10.89.3.33	GetTimestampLatchValue() const =0	679
10.89.3.34	GetTransferBlockID() const =0	679
10.89.3.35	GetTransferQueueCurrentBlockCount() const =0	679
10.89.3.36	GetWidth() const =0	680
10.89.3.37	SetChunks(GenApi::INodeMap &pNodeMap)=0	680
10.90	IDevFileStreamBase< CharType, Traits > Class Template Reference	680
10.90.1	Member Typedef Documentation	681
10.90.1.1	filebuf_type	681
10.90.1.2	ios_type	681
10.90.1.3	istream_type	681
10.90.2	Member Function Documentation	681
10.90.2.1	close()	681
10.90.2.2	is_open() const	681

10.90.2.3 open(Spinnaker::GenApi::INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::in) . . . . .	681
10.90.2.4 rdbuf() const . . . . .	682
10.91 IDevFileStreamBuf< CharType, Traits > Class Template Reference . . . . .	682
10.91.1 Constructor & Destructor Documentation . . . . .	683
10.91.1.1 IDevFileStreamBuf() . . . . .	683
10.91.1.2 ~IDevFileStreamBuf() . . . . .	683
10.91.2 Member Function Documentation . . . . .	683
10.91.2.1 close() . . . . .	683
10.91.2.2 is_open() const . . . . .	683
10.91.2.3 open(Spinnaker::GenApi::INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::in) . . . . .	683
10.91.2.4 pbackfail(int_type c) . . . . .	683
10.91.2.5 underflow() . . . . .	683
10.92 IDeviceEvent Class Reference . . . . .	683
10.92.1 Constructor & Destructor Documentation . . . . .	684
10.92.1.1 ~IDeviceEvent() . . . . .	684
10.92.1.2 IDeviceEvent() . . . . .	684
10.92.1.3 IDeviceEvent(const IDeviceEvent &) . . . . .	684
10.92.2 Member Function Documentation . . . . .	684
10.92.2.1 GetDeviceEventId() const =0 . . . . .	684
10.92.2.2 GetDeviceEventName() const =0 . . . . .	684
10.92.2.3 OnDeviceEvent(Spinnaker::GenICam::gcstring eventName)=0 . . . . .	685
10.92.2.4 operator=(const IDeviceEvent &) . . . . .	685
10.93 IImage Class Reference . . . . .	685
10.93.1 Detailed Description . . . . .	686
10.93.2 Constructor & Destructor Documentation . . . . .	687
10.93.2.1 ~IImage() . . . . .	687
10.93.2.2 IImage() . . . . .	687
10.93.3 Member Function Documentation . . . . .	687
10.93.3.1 CalculateStatistics(ImageStatistics &pStatistics)=0 . . . . .	687

10.93.3.2 CheckCRC() const =0 . . . . .	687
10.93.3.3 Convert(Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const =0 . . . . .	687
10.93.3.4 DeepCopy(const ImagePtr pSrcImage)=0 . . . . .	687
10.93.3.5 ExtractPolarization(const PolarizationAlgorithm polarizationAlogrithm, const PolarizationResolution resolution) const =0 . . . . .	687
10.93.3.6 GetBitsPerPixel() const =0 . . . . .	687
10.93.3.7 GetBufferSize() const =0 . . . . .	687
10.93.3.8 GetChunkData() const =0 . . . . .	687
10.93.3.9 GetChunkLayoutId() const =0 . . . . .	688
10.93.3.10GetColorProcessing() const =0 . . . . .	688
10.93.3.11GetData() const =0 . . . . .	688
10.93.3.12GetFrameId() const =0 . . . . .	688
10.93.3.13GetHeight() const =0 . . . . .	688
10.93.3.14GetID() const =0 . . . . .	688
10.93.3.15GetImageSize() const =0 . . . . .	688
10.93.3.16GetImageStatus() const =0 . . . . .	688
10.93.3.17GetNumChannels() const =0 . . . . .	688
10.93.3.18GetPayloadType() const =0 . . . . .	688
10.93.3.19GetPixelFormat() const =0 . . . . .	689
10.93.3.20GetPixelFormatIntType() const =0 . . . . .	689
10.93.3.21GetPixelFormatName() const =0 . . . . .	689
10.93.3.22GetPolarizationAlgorithm() const =0 . . . . .	689
10.93.3.23GetPolarizationValues() const =0 . . . . .	689
10.93.3.24GetPrivateData() const =0 . . . . .	689
10.93.3.25GetStride() const =0 . . . . .	689
10.93.3.26GetTimeStamp() const =0 . . . . .	689
10.93.3.27GetTLPayloadType() const =0 . . . . .	689
10.93.3.28GetTLPixelFormat() const =0 . . . . .	689
10.93.3.29GetTLPixelFormatNamespace() const =0 . . . . .	690
10.93.3.30GetValidPayloadSize() const =0 . . . . .	690



10.93.3.31GetWidth() const =0 . . . . .	690
10.93.3.32GetXOffset() const =0 . . . . .	690
10.93.3.33GetXPadding() const =0 . . . . .	690
10.93.3.34GetYOffset() const =0 . . . . .	690
10.93.3.35GetYPadding() const =0 . . . . .	690
10.93.3.36HasCRC() const =0 . . . . .	690
10.93.3.37IsIncomplete() const =0 . . . . .	690
10.93.3.38IsInUse()=0 . . . . .	690
10.93.3.39Release()=0 . . . . .	691
10.93.3.40ResetImage(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker← ::PixelFormatEnums pixelFormat)=0 . . . . .	691
10.93.3.41ResetImage(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker← ::PixelFormatEnums pixelFormat, void *pData)=0 . . . . .	691
10.93.3.42Save(const char *pFilename, ImageFileFormat format=FROM_FILE_EXT)=0 . . . . .	691
10.93.3.43Save(const char *pFilename, PNGOption &pOption)=0 . . . . .	691
10.93.3.44Save(const char *pFilename, PPMOption &pOption)=0 . . . . .	691
10.93.3.45Save(const char *pFilename, PGMOption &pOption)=0 . . . . .	691
10.93.3.46Save(const char *pFilename, TIFFOption &pOption)=0 . . . . .	691
10.93.3.47Save(const char *pFilename, JPEGOption &pOption)=0 . . . . .	691
10.93.3.48Save(const char *pFilename, JPG2Option &pOption)=0 . . . . .	691
10.93.3.49Save(const char *pFilename, BMPOption &pOption)=0 . . . . .	692
10.94 IImageEvent Class Reference . . . . .	692
10.94.1 Constructor & Destructor Documentation . . . . .	693
10.94.1.1 ~IImageEvent() . . . . .	693
10.94.1.2 IImageEvent() . . . . .	693
10.94.1.3 IImageEvent(const IImageEvent &) . . . . .	693
10.94.2 Member Function Documentation . . . . .	693
10.94.2.1 OnImageEvent(ImagePtr image)=0 . . . . .	693
10.94.2.2 operator=(const IImageEvent &) . . . . .	693
10.95 IImageStatistics Class Reference . . . . .	693
10.95.1 Detailed Description . . . . .	694

10.95.2 Constructor & Destructor Documentation . . . . .	694
10.95.2.1 ~IImageStatistics() . . . . .	694
10.95.2.2 IImageStatistics() . . . . .	694
10.95.2.3 IImageStatistics(const IImageStatistics &) . . . . .	694
10.95.3 Member Function Documentation . . . . .	694
10.95.3.1 DisableAll()=0 . . . . .	694
10.95.3.2 EnableAll()=0 . . . . .	694
10.95.3.3 EnableGreyOnly()=0 . . . . .	695
10.95.3.4 EnableHSLOnly()=0 . . . . .	695
10.95.3.5 EnableRGBOnly()=0 . . . . .	695
10.95.3.6 GetChannelStatus(StatisticsChannel channel, bool *pEnabled) const =0 . . . . .	695
10.95.3.7 GetHistogram(StatisticsChannel channel, int **ppHistogram) const =0 . . . . .	695
10.95.3.8 GetMean(StatisticsChannel channel, float *pPixelValueMean) const =0 . . . . .	695
10.95.3.9 GetNumPixelValues(StatisticsChannel channel, unsigned int *pNumPixelValues) const =0 . . . . .	695
10.95.3.10 GetPixelValueRange(StatisticsChannel channel, unsigned int *pPixelValueMin, unsigned int *pPixelValueMax) const =0 . . . . .	695
10.95.3.11 GetRange(StatisticsChannel channel, unsigned int *pMin, unsigned int *pMax) const =0 . . . . .	695
10.95.3.12 GetStatistics(StatisticsChannel channel, unsigned int *pRangeMin=NULL, un- signed int *pRangeMax=NULL, unsigned int *pPixelValueMin=NULL, unsigned int *pPixelValueMax=NULL, unsigned int *pNumPixelValues=NULL, float *p<- PixelValueMean=NULL, int **ppHistogram=NULL) const =0 . . . . .	696
10.95.3.13 SetChannelStatus(StatisticsChannel channel, bool enabled)=0 . . . . .	696
10.96 Interface Class Reference . . . . .	696
10.96.1 Detailed Description . . . . .	697
10.96.2 Constructor & Destructor Documentation . . . . .	697
10.96.2.1 ~IInterface() . . . . .	697
10.96.2.2 IInterface() . . . . .	697
10.96.2.3 IInterface(const IInterface &) . . . . .	697
10.96.3 Member Function Documentation . . . . .	697
10.96.3.1 GetCameras(bool updateCameras=true) const =0 . . . . .	697
10.96.3.2 GetTLNodeMap() const =0 . . . . .	698

10.96.3.3 IsInUse() const =0 . . . . .	698
10.96.3.4 operator=(const IInterface &) . . . . .	698
10.96.3.5 RegisterEvent(Event &evtToRegister)=0 . . . . .	698
10.96.3.6 SendActionCommand(unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL) const =0 . . . . .	698
10.96.3.7 UnregisterEvent(Event &evtToUnregister)=0 . . . . .	698
10.96.3.8 UpdateCameras()=0 . . . . .	698
10.96.4 Friends And Related Function Documentation . . . . .	698
10.96.4.1 InterfaceInternal . . . . .	698
10.96.5 Member Data Documentation . . . . .	698
10.96.5.1 m_pInterfaceData . . . . .	698
10.96.5.2 TLInterface . . . . .	698
10.97IInterfaceEvent Class Reference . . . . .	699
10.97.1 Constructor & Destructor Documentation . . . . .	700
10.97.1.1 ~IInterfaceEvent() . . . . .	700
10.97.1.2 IInterfaceEvent() . . . . .	700
10.97.1.3 IInterfaceEvent(const IInterfaceEvent &) . . . . .	700
10.97.2 Member Function Documentation . . . . .	700
10.97.2.1 OnDeviceArrival(uint64_t serialNumber)=0 . . . . .	700
10.97.2.2 OnDeviceRemoval(uint64_t serialNumber)=0 . . . . .	700
10.97.2.3 operator=(const IInterfaceEvent &) . . . . .	700
10.98IInterfaceList Class Reference . . . . .	700
10.98.1 Detailed Description . . . . .	701
10.98.2 Constructor & Destructor Documentation . . . . .	701
10.98.2.1 ~IInterfaceList(void) . . . . .	701
10.98.2.2 IInterfaceList(void) . . . . .	701
10.98.2.3 IInterfaceList(const IInterfaceList &) . . . . .	701
10.98.3 Member Function Documentation . . . . .	701
10.98.3.1 Clear()=0 . . . . .	701
10.98.3.2 GetByIndex(unsigned int index) const =0 . . . . .	701

10.98.3.3	GetSize() const =0 . . . . .	701
10.98.3.4	operator=(const IInterfaceList &) . . . . .	702
10.98.3.5	operator[](unsigned int index)=0 . . . . .	702
10.98.4	Member Data Documentation . . . . .	702
10.98.4.1	m_pInterfaceListData . . . . .	702
10.99	ILoggingEvent Class Reference . . . . .	702
10.99.1	Constructor & Destructor Documentation . . . . .	703
10.99.1.1	~ILoggingEvent() . . . . .	703
10.99.1.2	ILoggingEvent() . . . . .	703
10.99.1.3	ILoggingEvent(const ILoggingEvent &) . . . . .	703
10.99.2	Member Function Documentation . . . . .	703
10.99.2.1	OnLogEvent(LoggingEventDataPtr eventPtr)=0 . . . . .	703
10.99.2.2	operator=(const ILoggingEvent &) . . . . .	703
10.100	Image Class Reference . . . . .	704
10.100.1	Detailed Description . . . . .	707
10.100.2	Constructor & Destructor Documentation . . . . .	707
10.100.2.1	~Image() . . . . .	707
10.100.2.2	Image() . . . . .	708
10.100.2.3	Image(const ImagePtr image) . . . . .	708
10.100.2.4	Image(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::Pixel↵ FormatEnums pixelFormat, void *pData) . . . . .	708
10.100.3	Member Function Documentation . . . . .	708
10.100.3.1	CalculateStatistics(ImageStatistics &pStatistics) . . . . .	708
10.100.3.2	CheckCRC() const . . . . .	708
10.100.3.3	Convert(Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const . . . . .	708
10.100.3.4	Convert(Spinnaker::PixelFormatEnums format, Image &pDestImage, Color↵ ProcessingAlgorithm colorAlgorithm=DEFAULT) const . . . . .	709
10.100.3.5	Create() . . . . .	709
10.100.3.6	Create(const ImagePtr image) . . . . .	709
10.100.3.7	Create(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::↵ PixelFormatEnums pixelFormat, void *pData) . . . . .	709

10.100.3.8CreateShared() const . . . . .	709
10.100.3.9DeepCopy(const ImagePtr pSrcImage) . . . . .	709
10.100.3.10DeepCopy(const Image &pSrcImage) . . . . .	710
10.100.3.11ExtractPolarization(const PolarizationAlgorithm polarizationAlgorithm, const PolarizationResolution resolution) const . . . . .	710
10.100.3.12GetBitsPerPixel() const . . . . .	710
10.100.3.13GetBufferSize() const . . . . .	710
10.100.3.14GetChunkData() const . . . . .	711
10.100.3.15GetChunkLayoutId() const . . . . .	711
10.100.3.16GetColorProcessing() const . . . . .	711
10.100.3.17GetData() const . . . . .	711
10.100.3.18GetDefaultColorProcessing() . . . . .	712
10.100.3.19GetFrameID() const . . . . .	712
10.100.3.20GetHeatMapColorGradient(HeatMapColor &currentLowColor, HeatMapColor &currentHighColor) . . . . .	712
10.100.3.21GetHeatMapRange(unsigned int &currentLowValue, unsigned int &currentHigh← Value) . . . . .	712
10.100.3.22GetHeight() const . . . . .	713
10.100.3.23GetID() const . . . . .	713
10.100.3.24GetImageSize() const . . . . .	713
10.100.3.25GetImageStatus() const . . . . .	713
10.100.3.26GetImageStatusDescription(ImageStatus status) . . . . .	714
10.100.3.27GetNumChannels() const . . . . .	714
10.100.3.28GetPayloadType() const . . . . .	714
10.100.3.29GetPixelFormat() const . . . . .	714
10.100.3.30GetPixelFormatIntType() const . . . . .	715
10.100.3.31GetPixelFormatName() const . . . . .	715
10.100.3.32GetPolarizationAlgorithm() const . . . . .	715
10.100.3.33GetPolarizationValues() const . . . . .	715
10.100.3.34GetPrivateData() const . . . . .	716
10.100.3.35GetStride() const . . . . .	716
10.100.3.36GetTimeStamp() const . . . . .	716

10.100.3.37	GetTLPayloadType() const . . . . .	716
10.100.3.38	GetTLPixelFormat() const . . . . .	717
10.100.3.39	GetTLPixelFormatNamespace() const . . . . .	717
10.100.3.40	GetValidPayloadSize() const . . . . .	717
10.100.3.41	GetWidth() const . . . . .	718
10.100.3.42	GetXOffset() const . . . . .	718
10.100.3.43	GetXPadding() const . . . . .	718
10.100.3.44	GetYOffset() const . . . . .	718
10.100.3.45	GetYPadding() const . . . . .	719
10.100.3.46	HasCRC() const . . . . .	719
10.100.3.47	Incomplete() const . . . . .	719
10.100.3.48	InUse() . . . . .	719
10.100.3.49	Release() . . . . .	719
10.100.3.50	ResetImage(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker← ::PixelFormatEnums pixelFormat) . . . . .	719
10.100.3.51	ResetImage(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker← ::PixelFormatEnums pixelFormat, void *pData) . . . . .	720
10.100.3.52	Save(const char *pFilename, ImageFileFormat format=FROM_FILE_EXT) . . .	720
10.100.3.53	Save(const char *pFilename, PNGOption &pOption) . . . . .	720
10.100.3.54	Save(const char *pFilename, PPMOption &pOption) . . . . .	721
10.100.3.55	Save(const char *pFilename, PGMOption &pOption) . . . . .	721
10.100.3.56	Save(const char *pFilename, TIFFOption &pOption) . . . . .	721
10.100.3.57	Save(const char *pFilename, JPEGOption &pOption) . . . . .	721
10.100.3.58	Save(const char *pFilename, JPG2Option &pOption) . . . . .	722
10.100.3.59	Save(const char *pFilename, BMPOption &pOption) . . . . .	722
10.100.3.60	SetDefaultColorProcessing(ColorProcessingAlgorithm colorAlgorithm) . . . . .	722
10.100.3.61	SetHeatMapColorGradient(const HeatMapColor newLowColor, const HeatMap← Color newHighColor) . . . . .	722
10.100.3.62	SetHeatMapRange(const unsigned int newLowValue, const unsigned int new← HighValue) . . . . .	723
10.100.4	Friends And Related Function Documentation . . . . .	723
10.100.4.1	DataStream . . . . .	723

10.100.4.2ImageConverter . . . . .	723
10.100.4.3ImageFiler . . . . .	723
10.100.4.4ImageStatsCalculator . . . . .	723
10.100.4.5Stream . . . . .	723
10.100.5Member Data Documentation . . . . .	723
10.100.5.1m_pImageData . . . . .	723
10.101ImageEvent Class Reference . . . . .	724
10.101.1Detailed Description . . . . .	725
10.101.2Constructor & Destructor Documentation . . . . .	725
10.101.2.1ImageEvent() . . . . .	725
10.101.2.2~ImageEvent() . . . . .	725
10.101.3Member Function Documentation . . . . .	725
10.101.3.1OnImageEvent(ImagePtr image)=0 . . . . .	725
10.101.3.2operator=(const ImageEvent &) . . . . .	725
10.102ImagePtr Class Reference . . . . .	726
10.102.1Detailed Description . . . . .	727
10.102.2Constructor & Destructor Documentation . . . . .	727
10.102.2.1ImagePtr() . . . . .	727
10.102.2.2ImagePtr(const int) . . . . .	727
10.102.2.3~ImagePtr(void) . . . . .	727
10.102.3Member Function Documentation . . . . .	727
10.102.3.1operator=(const ImagePtr &) . . . . .	727
10.102.3.2operator=(const int nMustBeNull) . . . . .	727
10.103ImageStatistics Class Reference . . . . .	728
10.103.1Detailed Description . . . . .	729
10.103.2Constructor & Destructor Documentation . . . . .	729
10.103.2.1ImageStatistics() . . . . .	729
10.103.2.2~ImageStatistics() . . . . .	729
10.103.2.3ImageStatistics(const ImageStatistics &other) . . . . .	730
10.103.3Member Function Documentation . . . . .	730

10.103.3.1	<a href="#">DisableAll()</a> . . . . .	730
10.103.3.2	<a href="#">EnableAll()</a> . . . . .	730
10.103.3.3	<a href="#">EnableGreyOnly()</a> . . . . .	730
10.103.3.4	<a href="#">EnableHSLOnly()</a> . . . . .	730
10.103.3.5	<a href="#">EnableRGBOnly()</a> . . . . .	730
10.103.3.6	<a href="#">GetChannelStatus(StatisticsChannel channel, bool *pEnabled) const</a> . . . . .	730
10.103.3.7	<a href="#">GetHistogram(StatisticsChannel channel, int **ppHistogram) const</a> . . . . .	731
10.103.3.8	<a href="#">GetMean(StatisticsChannel channel, float *pPixelValueMean) const</a> . . . . .	731
10.103.3.9	<a href="#">GetNumPixelValues(StatisticsChannel channel, unsigned int *pNumPixelValues) const</a> . . . . .	731
10.103.3.10	<a href="#">GetPixelValueRange(StatisticsChannel channel, unsigned int *pPixelValueMin, unsigned int *pPixelValueMax) const</a> . . . . .	731
10.103.3.11	<a href="#">GetRange(StatisticsChannel channel, unsigned int *pMin, unsigned int *pMax) const</a> . . . . .	732
10.103.3.12	<a href="#">GetStatistics(StatisticsChannel channel, unsigned int *pRangeMin=NULL, unsigned int *pRangeMax=NULL, unsigned int *pPixelValueMin=NULL, unsigned int *pPixelValueMax=NULL, unsigned int *pNumPixelValues=NULL, float *pPixelValueMean=NULL, int **ppHistogram=NULL) const</a> . . . . .	732
10.103.3.13	<a href="#">operator=(const ImageStatistics &amp;other)</a> . . . . .	732
10.103.3.14	<a href="#">SetChannelStatus(StatisticsChannel channel, bool enabled)</a> . . . . .	733
10.103.4	<a href="#">Friends And Related Function Documentation</a> . . . . .	733
10.103.4.1	<a href="#">ImageStatsCalculator</a> . . . . .	733
10.104	<a href="#">int64_autovector_t Class Reference</a> . . . . .	733
10.104.1	<a href="#">Detailed Description</a> . . . . .	734
10.104.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	734
10.104.2.1	<a href="#">int64_autovector_t()</a> . . . . .	734
10.104.2.2	<a href="#">int64_autovector_t(const int64_autovector_t &amp;obj)</a> . . . . .	734
10.104.2.3	<a href="#">int64_autovector_t(size_t n)</a> . . . . .	734
10.104.2.4	<a href="#">~int64_autovector_t(void)</a> . . . . .	734
10.104.3	<a href="#">Member Function Documentation</a> . . . . .	734
10.104.3.1	<a href="#">operator delete(void *pWhere)</a> . . . . .	734
10.104.3.2	<a href="#">operator new(size_t uiSize)</a> . . . . .	734
10.104.3.3	<a href="#">operator=(const int64_autovector_t &amp;obj)</a> . . . . .	734



10.104.3.4	<code>operator[](size_t uiIndex)</code>	734
10.104.3.5	<code>operator[](size_t uiIndex) const</code>	734
10.104.3.6	<code>size() const</code>	734
10.104.4	Member Data Documentation	734
10.104.4.1	<code>pCount</code>	734
10.104.4.2	<code>pv</code>	734
10.105	IntegerNode Class Reference	735
10.105.1	Detailed Description	737
10.105.2	Constructor & Destructor Documentation	737
10.105.2.1	<code>IntegerNode()</code>	737
10.105.2.2	<code>IntegerNode(std::shared_ptr&lt; Node::NodeImpl &gt; pInteger)</code>	737
10.105.2.3	<code>~IntegerNode()</code>	737
10.105.3	Member Function Documentation	737
10.105.3.1	<code>GetFloatAlias()</code>	737
10.105.3.2	<code>GetInc()</code>	737
10.105.3.3	<code>GetIncMode()</code>	737
10.105.3.4	<code>GetListOfValidValues(bool bounded=true)</code>	737
10.105.3.5	<code>GetMax()</code>	738
10.105.3.6	<code>GetMin()</code>	738
10.105.3.7	<code>GetRepresentation()</code>	738
10.105.3.8	<code>GetUnit()</code>	738
10.105.3.9	<code>GetValue(bool Verify=false, bool IgnoreCache=false)</code>	738
10.105.3.10	<code>ImposeMax(int64_t Value)</code>	738
10.105.3.11	<code>ImposeMin(int64_t Value)</code>	738
10.105.3.12	<code>operator()()</code>	738
10.105.3.13	<code>operator*()</code>	739
10.105.3.14	<code>operator=(int64_t Value)</code>	739
10.105.3.15	<code>SetReference(INode *pBase)</code>	739
10.105.3.16	<code>SetValue(int64_t Value, bool Verify=true)</code>	739
10.106	Interface Class Reference	739

10.106.1Detailed Description . . . . .	740
10.106.2Constructor & Destructor Documentation . . . . .	741
10.106.2.1~Interface(void) . . . . .	741
10.106.3Member Function Documentation . . . . .	741
10.106.3.1GetCameras(bool updateCameras=true) const . . . . .	741
10.106.3.2GetTLNodeMap() const . . . . .	741
10.106.3.3IsInUse() const . . . . .	741
10.106.3.4RegisterEvent(Event &evtToRegister) . . . . .	742
10.106.3.5SendActionCommand(unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL) const . . . . .	742
10.106.3.6UnregisterEvent(Event &evtToUnregister) . . . . .	742
10.106.3.7UpdateCameras() . . . . .	742
10.106.4Friends And Related Function Documentation . . . . .	743
10.106.4.1InterfaceInternal . . . . .	743
10.107InterfaceEvent Class Reference . . . . .	743
10.107.1Detailed Description . . . . .	744
10.107.2Constructor & Destructor Documentation . . . . .	745
10.107.2.1InterfaceEvent() . . . . .	745
10.107.2.2~InterfaceEvent() . . . . .	745
10.107.3Member Function Documentation . . . . .	745
10.107.3.1OnDeviceArrival(uint64_t serialNumber)=0 . . . . .	745
10.107.3.2OnDeviceRemoval(uint64_t serialNumber)=0 . . . . .	745
10.107.3.3operator=(const InterfaceEvent &) . . . . .	745
10.108InterfaceList Class Reference . . . . .	745
10.108.1Detailed Description . . . . .	747
10.108.2Constructor & Destructor Documentation . . . . .	747
10.108.2.1InterfaceList(void) . . . . .	747
10.108.2.2~InterfaceList(void) . . . . .	747
10.108.2.3InterfaceList(const InterfaceList &iface) . . . . .	747
10.108.3Member Function Documentation . . . . .	747

10.108.3.1Clear()	747
10.108.3.2GetByIndex(unsigned int index) const	747
10.108.3.3GetSize() const	747
10.108.3.4operator=(const InterfaceList &iface)	748
10.108.3.5operator[](unsigned int index)	748
10.108.4Friends And Related Function Documentation	748
10.108.4.1SystemImpl	748
10.109InterfacePtr Class Reference	748
10.109.1Detailed Description	749
10.109.2Constructor & Destructor Documentation	749
10.109.2.1InterfacePtr()	749
10.109.2.2InterfacePtr(const int)	749
10.109.2.3~InterfacePtr(void)	749
10.109.3Member Function Documentation	749
10.109.3.1operator=(const int nMustBeNull)	749
10.110IntRegNode Class Reference	750
10.110.1Detailed Description	751
10.110.2Constructor & Destructor Documentation	751
10.110.2.1IntRegNode()	751
10.110.2.2IntRegNode(std::shared_ptr< Node::NodeImpl > pInteger)	751
10.110.2.3~IntRegNode()	752
10.110.3Member Function Documentation	752
10.110.3.1SetReference(INode *pBase)	752
10.111IRemovalEvent Class Reference	752
10.111.1Constructor & Destructor Documentation	753
10.111.1.1~IRemovalEvent()	753
10.111.1.2RemovalEvent()	753
10.111.1.3RemovalEvent(const IRemovalEvent &)	753
10.111.2Member Function Documentation	753
10.111.2.1OnDeviceRemoval(uint64_t serialNumber)=0	753

10.111.2.2operator=(const IRemovalEvent &)	753
10.112System Class Reference	753
10.112.1Detailed Description	754
10.112.2Constructor & Destructor Documentation	754
10.112.2.1~ISystem()	754
10.112.2.2ISystem()	754
10.112.2.3ISystem(const ISystem &)	754
10.112.3Member Function Documentation	754
10.112.3.1GetCameras(bool updateInterfaces=true, bool updateCameras=true)=0	754
10.112.3.2GetInterfaces(bool updateInterface=true)=0	754
10.112.3.3GetLibraryVersion()=0	755
10.112.3.4GetLoggingEventPriorityLevel()=0	755
10.112.3.5IsInUse()=0	755
10.112.3.6operator=(const ISystem &)	755
10.112.3.7RegisterInterfaceEvent(Event &evtToRegister, bool updateInterface=true)=0	755
10.112.3.8RegisterLoggingEvent(LoggingEvent &handler)=0	755
10.112.3.9ReleaseInstance()=0	755
10.112.3.10SendActionCommand(unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL)=0	755
10.112.3.11SetLoggingEventPriorityLevel(SpinnakerLogLevel level)=0	755
10.112.3.12UnregisterAllLoggingEvent()=0	755
10.112.3.13UnregisterInterfaceEvent(Event &evtToUnregister)=0	756
10.112.3.14UnregisterLoggingEvent(LoggingEvent &handler)=0	756
10.112.3.15UpdateCameras(bool updateInterfaces=true)=0	756
10.113PEGOOption Struct Reference	756
10.113.1Detailed Description	756
10.113.2Constructor & Destructor Documentation	756
10.113.2.1JPEGOOption()	756
10.113.3Member Data Documentation	756
10.113.3.1progressive	756

10.113.3.2quality	757
10.113.3.3reserved	757
10.114PG2Option Struct Reference	757
10.114.1Detailed Description	757
10.114.2Constructor & Destructor Documentation	757
10.114.2.1JPG2Option()	757
10.114.3Member Data Documentation	757
10.114.3.1quality	757
10.114.3.2reserved	758
10.115LibraryVersion Struct Reference	758
10.115.1Detailed Description	758
10.115.2Member Data Documentation	758
10.115.2.1build	758
10.115.2.2major	758
10.115.2.3minor	758
10.115.2.4type	759
10.116LockableObject< Object >::Lock Class Reference	759
10.116.1Detailed Description	759
10.116.2Constructor & Destructor Documentation	759
10.116.2.1Lock(const LockableObject< Object > &obj)	759
10.116.2.2~Lock()	759
10.117LockableObject< Object > Class Template Reference	760
10.117.1Detailed Description	760
10.117.2Member Function Documentation	761
10.117.2.1GetLock() const	761
10.117.3Friends And Related Function Documentation	761
10.117.3.1Lock	761
10.117.4Member Data Documentation	761
10.117.4.1m_Lock	761
10.118LoggingEvent Class Reference	761

10.118.1Detailed Description . . . . .	762
10.118.2Constructor & Destructor Documentation . . . . .	762
10.118.2.1LoggingEvent() . . . . .	762
10.118.2.2~LoggingEvent() . . . . .	763
10.118.3Member Function Documentation . . . . .	763
10.118.3.1OnLogEvent(LoggingEventDataPtr eventPtr)=0 . . . . .	763
10.118.3.2operator=(const LoggingEvent &) . . . . .	763
10.119LoggingEventData Class Reference . . . . .	763
10.119.1Detailed Description . . . . .	764
10.119.2Constructor & Destructor Documentation . . . . .	764
10.119.2.1~LoggingEventData() . . . . .	764
10.119.2.2LoggingEventData(void *data) . . . . .	764
10.119.3Member Function Documentation . . . . .	764
10.119.3.1GetCategoryName() . . . . .	764
10.119.3.2GetLogMessage() . . . . .	764
10.119.3.3GetNDC() . . . . .	765
10.119.3.4GetPriority() . . . . .	765
10.119.3.5GetPriorityName() . . . . .	765
10.119.3.6GetThreadName() . . . . .	765
10.119.3.7GetTimestamp() . . . . .	765
10.119.4Friends And Related Function Documentation . . . . .	765
10.119.4.1SystemImpl . . . . .	765
10.120LoggingEventDataPtr Class Reference . . . . .	766
10.120.1Detailed Description . . . . .	766
10.120.2Constructor & Destructor Documentation . . . . .	767
10.120.2.1LoggingEventDataPtr() . . . . .	767
10.120.2.2LoggingEventDataPtr(const int) . . . . .	767
10.120.2.3~LoggingEventDataPtr(void) . . . . .	767
10.120.3Member Function Documentation . . . . .	767
10.120.3.1operator=(const int nMustBeNull) . . . . .	767

10.12	<del>Member_NodeCallback</del> < Client, Member > Class Template Reference . . . . .	767
10.121.	<del>1</del> Detailed Description . . . . .	768
10.121.	<del>2</del> Member Typedef Documentation . . . . .	768
10.121.2.	<del>1</del> PMEMBERFUNC . . . . .	768
10.121.	<del>3</del> Constructor & Destructor Documentation . . . . .	769
10.121.3.	<del>1</del> Member_NodeCallback(INode *pNode, Client &client, Member member, E↔ CallbackType CallbackType) . . . . .	769
10.121.	<del>4</del> Member Function Documentation . . . . .	769
10.121.4.	<del>1</del> Destroy() . . . . .	769
10.121.4.	<del>2</del> operator()(ECallbackType CallbackType) const . . . . .	769
10.12	<del>M</del> MJPGOption Struct Reference . . . . .	769
10.122.	<del>1</del> Detailed Description . . . . .	769
10.122.	<del>2</del> Constructor & Destructor Documentation . . . . .	770
10.122.2.	<del>1</del> MJPGOption() . . . . .	770
10.122.	<del>3</del> Member Data Documentation . . . . .	770
10.122.3.	<del>1</del> frameRate . . . . .	770
10.122.3.	<del>2</del> quality . . . . .	770
10.122.3.	<del>3</del> reserved . . . . .	770
10.12	<del>N</del> Node Class Reference . . . . .	770
10.123.	<del>1</del> Detailed Description . . . . .	773
10.123.	<del>2</del> Constructor & Destructor Documentation . . . . .	773
10.123.2.	<del>1</del> Node() . . . . .	773
10.123.2.	<del>2</del> Node(std::shared_ptr< Node::NodeImpl > pNodeHandle) . . . . .	773
10.123.2.	<del>3</del> ~Node() . . . . .	773
10.123.	<del>3</del> Member Function Documentation . . . . .	773
10.123.3.	<del>1</del> DeregisterCallback(CallbackHandleType hCallback) . . . . .	773
10.123.3.	<del>2</del> GetAccessMode() const . . . . .	773
10.123.3.	<del>3</del> GetAlias() const . . . . .	774
10.123.3.	<del>4</del> GetCachingMode() const . . . . .	774
10.123.3.	<del>5</del> GetCastAlias() const . . . . .	774

10.123.3.6GetChildren(GenApi::NodeList_t &Children, ELinkType LinkType=ctReading↵ Children) const . . . . .	774
10.123.3.7GetDescription() const . . . . .	774
10.123.3.8GetDeviceName() const . . . . .	774
10.123.3.9GetDisplayName() const . . . . .	774
10.123.3.10GetDocuURL() const . . . . .	774
10.123.3.11GetEventID() const . . . . .	774
10.123.3.12GetName(bool FullQualified=false) const . . . . .	775
10.123.3.13GetNameSpace() const . . . . .	775
10.123.3.14GetNodeHandle() const . . . . .	775
10.123.3.15GetNodeMap() const . . . . .	775
10.123.3.16GetParents(GenApi::NodeList_t &Parents) const . . . . .	775
10.123.3.17GetPollingTime() const . . . . .	775
10.123.3.18GetPrincipalInterfaceType() const . . . . .	775
10.123.3.19GetProperty(const GenICam::gcstring &PropertyName, GenICam::gcstring &↵ ValueStr, GenICam::gcstring &AttributeStr) . . . . .	775
10.123.3.20GetPropertyNames(GenICam::gcstring_vector &PropertyNames) const . . . . .	775
10.123.3.21GetSelectedFeatures(FeatureList_t &) const . . . . .	776
10.123.3.22GetSelectingFeatures(FeatureList_t &) const . . . . .	776
10.123.3.23GetToolTip() const . . . . .	776
10.123.3.24GetVisibility() const . . . . .	776
10.123.3.25ImposeAccessMode(EAccessMode ImposedAccessMode) . . . . .	776
10.123.3.26ImposeVisibility(EVisibility ImposedVisibility) . . . . .	776
10.123.3.27validateNode() . . . . .	776
10.123.3.28AccessModeCacheable() const . . . . .	776
10.123.3.29Cachable() const . . . . .	776
10.123.3.30Deprecated() const . . . . .	776
10.123.3.31Feature() const . . . . .	777
10.123.3.32Selector() const . . . . .	777
10.123.3.33Streamable() const . . . . .	777
10.123.3.34operator!=(int nullPtr) const . . . . .	777



10.123.3.35	operator==(int nullPtr) const . . . . .	777
10.123.3.36	RegisterCallback(CNodeCallback *pCallback) . . . . .	777
10.123.3.37	SetNodeHandle(std::shared_ptr< Node::NodeImpl > pNodeHandle) . . . . .	777
10.123.3.38	SetNodeMap(INodeMap *pNodeMap) . . . . .	777
10.123.3.39	SetReference(INode *pBase) . . . . .	777
10.123.3.40	SetReference(ISelector *pBase) . . . . .	777
10.123.4	Member Data Documentation . . . . .	777
10.123.4.1	m_Callbacks . . . . .	777
10.123.4.2	m_pNodeData . . . . .	778
10.123.4.3	m_pNodeMap . . . . .	778
10.124	NodeMap Class Reference . . . . .	778
10.124.1	Detailed Description . . . . .	780
10.124.2	Constructor & Destructor Documentation . . . . .	780
10.124.2.1	NodeMap(GenICam::gcstring DeviceName=""Device"") . . . . .	780
10.124.2.2	~NodeMap() . . . . .	780
10.124.3	Member Function Documentation . . . . .	780
10.124.3.1	ClearXMLCache() . . . . .	780
10.124.3.2	Connect(IPort *pPort, const GenICam::gcstring &PortName) const . . . . .	781
10.124.3.3	Connect(IPort *pPort) const . . . . .	781
10.124.3.4	Destroy() . . . . .	781
10.124.3.5	GetDeviceName() . . . . .	781
10.124.3.6	GetDeviceVersion(GenICam::Version_t &Version) . . . . .	781
10.124.3.7	GetGenApiVersion(GenICam::Version_t &Version, uint16_t &Build) . . . . .	781
10.124.3.8	GetLock() const . . . . .	781
10.124.3.9	GetModelName() . . . . .	781
10.124.3.10	GetNode(const GenICam::gcstring &key) const . . . . .	781
10.124.3.11	GetNodeMapHandle() const . . . . .	781
10.124.3.12	GetNodes(NodeList_t &Nodes) const . . . . .	781
10.124.3.13	GetNumNodes() const . . . . .	782
10.124.3.14	GetProductGuid() . . . . .	782

10.124.3.16	GetSchemaVersion(GenICam::Version_t &Version) . . . . .	782
10.124.3.16	GetStandardNameSpace() . . . . .	782
10.124.3.17	GetSupportedSchemaVersions(GenICam::gcstring_vector &SchemaVersions) . . . . .	782
10.124.3.18	GetToolTip() . . . . .	782
10.124.3.19	GetVendorName() . . . . .	783
10.124.3.20	GetVersionGuid() . . . . .	783
10.124.3.21	InvalidateNodes() const . . . . .	783
10.124.3.22	LoadXMLFromFile(GenICam::gcstring FileName) . . . . .	783
10.124.3.23	LoadXMLFromFileInject(GenICam::gcstring TargetFileName, GenICam::gcstring InjectFileName) . . . . .	783
10.124.3.24	LoadXMLFromString(const GenICam::gcstring &XMLData) . . . . .	783
10.124.3.25	LoadXMLFromStringInject(const GenICam::gcstring &TargetXMLDataconst, const GenICam::gcstring &InjectXMLData) . . . . .	783
10.124.3.26	LoadXMLFromZIPData(const void *zipData, size_t zipSize) . . . . .	783
10.124.3.27	LoadXMLFromZIPFile(GenICam::gcstring ZipFileName) . . . . .	783
10.124.3.28	Poll(int64_t ElapsedTime) . . . . .	784
10.124.4	Member Data Documentation . . . . .	784
10.124.4.1	_Ptr . . . . .	784
10.125	NodeMapFactory::NodeStatistics_t Struct Reference . . . . .	784
10.125.1	Member Data Documentation . . . . .	784
10.125.1.1	NumLinks . . . . .	784
10.125.1.2	NumNodes . . . . .	784
10.125.1.3	NumProperties . . . . .	784
10.125.1.4	NumStrings . . . . .	784
10.126	DevFileStreamBase< CharType, Traits > Class Template Reference . . . . .	785
10.126.1	Member Typedef Documentation . . . . .	786
10.126.1.1	filebuf_type . . . . .	786
10.126.1.2	os_type . . . . .	786
10.126.1.3	ostream_type . . . . .	786
10.126.2	Member Function Documentation . . . . .	786
10.126.2.1	close() . . . . .	786

10.126.2.2s_open() const . . . . .	786
10.126.2.3open(INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::out"   std::ios_base::trunc) . . . . .	786
10.126.2.4dbuf() const . . . . .	786
10.127.0DevFileStreamBuf< CharType, Traits > Class Template Reference . . . . .	787
10.127.1Constructor & Destructor Documentation . . . . .	788
10.127.1.1ODevFileStreamBuf() . . . . .	788
10.127.1.2~ODevFileStreamBuf() . . . . .	788
10.127.2Member Function Documentation . . . . .	788
10.127.2.1close() . . . . .	788
10.127.2.2s_open() const . . . . .	788
10.127.2.3open(Spinnaker::GenApi::INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode) . . . . .	788
10.127.2.4overflow(int_type c=traits_type::eof()) . . . . .	788
10.127.2.5sync() . . . . .	788
10.127.2.6sputn(const char_type *s, std::streamsize n) . . . . .	788
10.128.0GMOption Struct Reference . . . . .	788
10.128.1Detailed Description . . . . .	788
10.128.2Constructor & Destructor Documentation . . . . .	789
10.128.2.1PGMOption() . . . . .	789
10.128.3Member Data Documentation . . . . .	789
10.128.3.1binaryFile . . . . .	789
10.128.3.2reserved . . . . .	789
10.129.0PNGOption Struct Reference . . . . .	789
10.129.1Detailed Description . . . . .	789
10.129.2Constructor & Destructor Documentation . . . . .	790
10.129.2.1PNGOption() . . . . .	790
10.129.3Member Data Documentation . . . . .	790
10.129.3.1compressionLevel . . . . .	790
10.129.3.2interlaced . . . . .	790
10.129.3.3reserved . . . . .	790

10.130. <del>PortNode</del> Class Reference . . . . .	790
10.130.1. Detailed Description . . . . .	792
10.130.2. Constructor & Destructor Documentation . . . . .	792
10.130.2.1. <del>PortNode()</del> . . . . .	792
10.130.2.2. <del>PortNode(std::shared_ptr&lt; Node::NodeImpl &gt; pValue)</del> . . . . .	792
10.130.2.3. <del>~PortNode()</del> . . . . .	792
10.130.3. Member Function Documentation . . . . .	792
10.130.3.1. <del>CacheChunkData()</del> const . . . . .	792
10.130.3.2. <del>GetChunkID()</del> const . . . . .	792
10.130.3.3. <del>GetPortHandle()</del> . . . . .	792
10.130.3.4. <del>GetSwapEndianness()</del> . . . . .	792
10.130.3.5. <del>Read(void *pBuffer, int64_t Address, int64_t Length)</del> . . . . .	792
10.130.3.6. <del>Replay(IPortWriteList *pPortRecorder, bool Invalidate=true)</del> . . . . .	793
10.130.3.7. <del>SetPortImpl(IPort *pPort)</del> . . . . .	793
10.130.3.8. <del>SetReference(INode *pBase)</del> . . . . .	793
10.130.3.9. <del>SetReference(IPort *pBase)</del> . . . . .	793
10.130.3.10. <del>SetReference(IChunkPort *pBase)</del> . . . . .	793
10.130.3.11. <del>StartRecording(IPortWriteList *pPortRecorder)</del> . . . . .	793
10.130.3.12. <del>StopRecording()</del> . . . . .	793
10.130.3.13. <del>Write(const void *pBuffer, int64_t Address, int64_t Length)</del> . . . . .	794
10.131. <del>PortRecorder</del> Class Reference . . . . .	794
10.131.1. Detailed Description . . . . .	795
10.131.2. Constructor & Destructor Documentation . . . . .	796
10.131.2.1. <del>PortRecorder()</del> . . . . .	796
10.131.2.2. <del>~PortRecorder()</del> . . . . .	796
10.131.3. Member Function Documentation . . . . .	796
10.131.3.1. <del>GetAccessMode()</del> const . . . . .	796
10.131.3.2. <del>SetReference(IPort *pBase)</del> . . . . .	796
10.131.3.3. <del>StartRecording(IPortWriteList *pPortRecorder)</del> . . . . .	796
10.131.3.4. <del>StopRecording()</del> . . . . .	796

10.132	PortReplay Class Reference	797
10.132.1	Detailed Description	798
10.132.2	Constructor & Destructor Documentation	798
10.132.2.1	PortReplay()	798
10.132.2.2	~PortReplay()	798
10.132.3	Member Function Documentation	798
10.132.3.1	GetPortReplayHandle()	798
10.132.3.2	Replay(IPortWriteList *pPortRecorder, bool Invalidate=true)	798
10.132.3.3	SetReference(IPort *pBase)	798
10.133	PPMOption Struct Reference	799
10.133.1	Detailed Description	799
10.133.2	Constructor & Destructor Documentation	799
10.133.2.1	PPMOption()	799
10.133.3	Member Data Documentation	799
10.133.3.1	binaryFile	799
10.133.3.2	reserved	799
10.134	RegisterNode Class Reference	800
10.134.1	Detailed Description	801
10.134.2	Constructor & Destructor Documentation	802
10.134.2.1	RegisterNode()	802
10.134.2.2	RegisterNode(std::shared_ptr< Node::NodeImpl > pRegister)	802
10.134.2.3	~RegisterNode()	802
10.134.3	Member Function Documentation	802
10.134.3.1	Get(uint8_t *pBuffer, int64_t Length, bool Verify=false, bool IgnoreCache=false)	802
10.134.3.2	GetAddress()	802
10.134.3.3	GetLength()	802
10.134.3.4	Set(const uint8_t *pBuffer, int64_t Length, bool Verify=true)	802
10.134.3.5	SetReference(INode *pBase)	803
10.135	RemovalEvent Class Reference	803
10.135.1	Detailed Description	804

10.135.2	Constructor & Destructor Documentation . . . . .	804
10.135.2.1	RemovalEvent() . . . . .	804
10.135.2.2	~RemovalEvent() . . . . .	804
10.135.3	Member Function Documentation . . . . .	804
10.135.3.1	OnDeviceRemoval(uint64_t serialNumber)=0 . . . . .	804
10.135.3.2	operator=(const RemovalEvent &) . . . . .	805
10.136	SingleChunkData_t Struct Reference . . . . .	805
10.136.1	Member Data Documentation . . . . .	805
10.136.1.1	ChunkID . . . . .	805
10.136.1.2	ChunkLength . . . . .	805
10.136.1.3	ChunkOffset . . . . .	805
10.137	SingleChunkDataStr_t Struct Reference . . . . .	805
10.137.1	Member Data Documentation . . . . .	805
10.137.1.1	ChunkID . . . . .	805
10.137.1.2	ChunkLength . . . . .	805
10.137.1.3	ChunkOffset . . . . .	805
10.138	SpinTestCamera Class Reference . . . . .	806
10.139	SpinVideo Class Reference . . . . .	806
10.139.1	Detailed Description . . . . .	807
10.139.2	Constructor & Destructor Documentation . . . . .	807
10.139.2.1	SpinVideo() . . . . .	807
10.139.2.2	~SpinVideo() . . . . .	807
10.139.3	Member Function Documentation . . . . .	807
10.139.3.1	Append(ImagePtr pImage) . . . . .	807
10.139.3.2	Close() . . . . .	808
10.139.3.3	Open(const char *pFileName, AVIOption &pOption) . . . . .	808
10.139.3.4	Open(const char *pFileName, MJPGOption &pOption) . . . . .	808
10.139.3.5	Open(const char *pFileName, H264Option &pOption) . . . . .	808
10.139.3.6	SetMaximumFileSize(unsigned int size) . . . . .	809
10.140	StringNode Class Reference . . . . .	809

10.140.1Detailed Description . . . . .	811
10.140.2Constructor & Destructor Documentation . . . . .	811
10.140.2.1StringNode() . . . . .	811
10.140.2.2StringNode(std::shared_ptr< Node::NodeImpl > pString) . . . . .	811
10.140.2.3~StringNode() . . . . .	811
10.140.3Member Function Documentation . . . . .	811
10.140.3.1GetMaxLength() . . . . .	811
10.140.3.2GetValue(bool Verify=false, bool IgnoreCache=false) . . . . .	811
10.140.3.3operator()() . . . . .	812
10.140.3.4operator*() . . . . .	812
10.140.3.5operator=(const GenICam::gcstring &Value) . . . . .	812
10.140.3.6SetReference(INode *pBase) . . . . .	812
10.140.3.7SetValue(const GenICam::gcstring &Value, bool Verify=true) . . . . .	812
10.141StringRegNode Class Reference . . . . .	813
10.141.1Detailed Description . . . . .	814
10.141.2Constructor & Destructor Documentation . . . . .	814
10.141.2.1StringRegNode() . . . . .	814
10.141.2.2StringRegNode(std::shared_ptr< Node::NodeImpl > pString) . . . . .	814
10.141.2.3~StringRegNode() . . . . .	815
10.141.3Member Function Documentation . . . . .	815
10.141.3.1SetReference(INode *pBase) . . . . .	815
10.142System Class Reference . . . . .	815
10.142.1Detailed Description . . . . .	816
10.142.2Constructor & Destructor Documentation . . . . .	817
10.142.2.1~System() . . . . .	817
10.142.2.2System() . . . . .	817
10.142.3Member Function Documentation . . . . .	817
10.142.3.1GetCameras(bool updateInterfaces=true, bool updateCameras=true) . . . . .	817
10.142.3.2GetInstance() . . . . .	817
10.142.3.3GetInterfaces(bool updateInterface=true) . . . . .	818

10.142.3.4	GetLibraryVersion()	818
10.142.3.5	GetLoggingEventPriorityLevel()	818
10.142.3.6	IsInUse()	819
10.142.3.7	RegisterInterfaceEvent(Event &evtToRegister, bool updateInterface=true)	819
10.142.3.8	RegisterLoggingEvent(LoggingEvent &handler)	819
10.142.3.9	ReleaseInstance()	819
10.142.3.10	SendActionCommand(unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL)	820
10.142.3.11	SetLoggingEventPriorityLevel(SpinnakerLogLevel level)	820
10.142.3.12	UnregisterAllLoggingEvent()	821
10.142.3.13	UnregisterInterfaceEvent(Event &evtToUnregister)	821
10.142.3.14	UnregisterLoggingEvent(LoggingEvent &handler)	821
10.142.3.15	UpdateCameras(bool updateInterfaces=true)	821
10.143	SystemPtr Class Reference	822
10.143.1	Detailed Description	823
10.143.2	Constructor & Destructor Documentation	823
10.143.2.1	SystemPtr()	823
10.143.2.2	SystemPtr(const int)	823
10.143.2.3	~SystemPtr(void)	823
10.144	TIFFOption Struct Reference	823
10.144.1	Detailed Description	824
10.144.2	Member Enumeration Documentation	824
10.144.2.1	CompressionMethod	824
10.144.3	Constructor & Destructor Documentation	824
10.144.3.1	TIFFOption()	824
10.144.4	Member Data Documentation	824
10.144.4.1	compression	824
10.144.4.2	reserved	824
10.145	TransportLayerDevice Class Reference	825
10.145.1	Detailed Description	826



10.145.2	Constructor & Destructor Documentation . . . . .	827
10.145.2.1	TransportLayerDevice(GenApi::INodeMap *nodeMapTLDevice) . . . . .	827
10.145.2.2	~TransportLayerDevice() . . . . .	827
10.145.2.3	TransportLayerDevice() . . . . .	827
10.145.3	Friends And Related Function Documentation . . . . .	827
10.145.3.1	CameraBase . . . . .	827
10.145.3.2	CameraInternal . . . . .	827
10.145.3.3	CameraBase . . . . .	827
10.145.4	Member Data Documentation . . . . .	827
10.145.4.1	DeviceAccessStatus . . . . .	827
10.145.4.2	DeviceCurrentSpeed . . . . .	827
10.145.4.3	DeviceDisplayName . . . . .	827
10.145.4.4	DeviceDriverVersion . . . . .	827
10.145.4.5	DeviceEndiannessMechanism . . . . .	827
10.145.4.6	DeviceID . . . . .	828
10.145.4.7	DeviceInstanceld . . . . .	828
10.145.4.8	DeviceIsUpdater . . . . .	828
10.145.4.9	DeviceLinkSpeed . . . . .	828
10.145.4.10	DeviceModelName . . . . .	828
10.145.4.11	DeviceMulticastMonitorMode . . . . .	828
10.145.4.12	DeviceSerialNumber . . . . .	828
10.145.4.13	DeviceType . . . . .	828
10.145.4.14	DeviceU3VProtocol . . . . .	829
10.145.4.15	DeviceUserID . . . . .	829
10.145.4.16	DeviceVendorName . . . . .	829
10.145.4.17	DeviceVersion . . . . .	829
10.145.4.18	GenICamXMLLocation . . . . .	829
10.145.4.19	GenICamXMLPath . . . . .	829
10.145.4.20	DevCCP . . . . .	829
10.145.4.21	DevDeviceDiscoverMaximumPacketSize . . . . .	829

10.145.4.22	evDeviceGateway	830
10.145.4.23	evDeviceIPAddress	830
10.145.4.24	evDevicesWrongSubnet	830
10.145.4.25	evDeviceMACAddress	830
10.145.4.26	evDeviceMaximumPacketSize	830
10.145.4.27	evDeviceMaximumRetryCount	830
10.145.4.28	evDeviceModelsBigEndian	830
10.145.4.29	evDevicePort	831
10.145.4.30	evDeviceReadAndWriteTimeout	831
10.145.4.31	evDeviceSubnetMask	831
10.145.4.32	evVersionMajor	831
10.145.4.33	evVersionMinor	831
10.145.4.34	UIXMLLocation	831
10.145.4.35	UIXMLPath	831
10.146	TransportLayerInterface Class Reference	832
10.146.1	Detailed Description	834
10.146.2	Constructor & Destructor Documentation	834
10.146.2.1	TransportLayerInterface(GenApi::INodeMap *nodeMapTLDevice)	834
10.146.2.2	~TransportLayerInterface()	834
10.146.2.3	TransportLayerInterface()	834
10.146.3	Friends And Related Function Documentation	834
10.146.3.1	Interface	834
10.146.3.2	Interface	834
10.146.3.3	InterfaceInternal	834
10.146.4	Member Data Documentation	834
10.146.4.1	ActionCommand	834
10.146.4.2	AutoForceIP	834
10.146.4.3	DeviceAccessStatus	834
10.146.4.4	DeviceCount	834
10.146.4.5	DeviceID	835

10.146.4.6DeviceModelName . . . . .	835
10.146.4.7DeviceSelector . . . . .	835
10.146.4.8DeviceUnlock . . . . .	835
10.146.4.9DeviceUpdateList . . . . .	835
10.146.4.10DeviceVendorName . . . . .	835
10.146.4.11GevActionDeviceKey . . . . .	835
10.146.4.12GevActionGroupKey . . . . .	836
10.146.4.13GevActionGroupMask . . . . .	836
10.146.4.14GevActionTime . . . . .	836
10.146.4.15GevDeviceIPAddress . . . . .	836
10.146.4.16GevDeviceMACAddress . . . . .	836
10.146.4.17GevDeviceSubnetMask . . . . .	836
10.146.4.18GevInterfaceGateway . . . . .	836
10.146.4.19GevInterfaceIPAddress . . . . .	837
10.146.4.20GevInterfaceMACAddress . . . . .	837
10.146.4.21GevInterfaceSubnetMask . . . . .	837
10.146.4.22HostAdapterDriverVersion . . . . .	837
10.146.4.23HostAdapterName . . . . .	837
10.146.4.24HostAdapterVendor . . . . .	837
10.146.4.25compatibleDeviceCount . . . . .	837
10.146.4.26compatibleDeviceID . . . . .	837
10.146.4.27compatibleDeviceModelName . . . . .	838
10.146.4.28compatibleDeviceSelector . . . . .	838
10.146.4.29compatibleDeviceVendorName . . . . .	838
10.146.4.30compatibleGevDeviceIPAddress . . . . .	838
10.146.4.31compatibleGevDeviceMACAddress . . . . .	838
10.146.4.32compatibleGevDeviceSubnetMask . . . . .	838
10.146.4.33InterfaceDisplayName . . . . .	838
10.146.4.34InterfaceID . . . . .	839
10.146.4.35InterfaceType . . . . .	839

10.146.4.3	BOEStatus	839
10.147	TransportLayerStream Class Reference	839
10.147.1	Detailed Description	840
10.147.2	Constructor & Destructor Documentation	841
10.147.2.1	TransportLayerStream(GenApi::INodeMap *nodeMapTLDevice)	841
10.147.2.2	~TransportLayerStream()	841
10.147.2.3	TransportLayerStream()	841
10.147.3	Friends And Related Function Documentation	841
10.147.3.1	CameraBase	841
10.147.3.2	CameraInternal	841
10.147.3.3	CameraBase	841
10.147.4	Member Data Documentation	841
10.147.4.1	GevFailedPacketCount	841
10.147.4.2	GevMaximumNumberResendBuffers	841
10.147.4.3	GevMaximumNumberResendRequests	841
10.147.4.4	GevPacketResendMode	841
10.147.4.5	GevPacketResendTimeout	841
10.147.4.6	GevResendPacketCount	842
10.147.4.7	GevResendRequestCount	842
10.147.4.8	GevTotalPacketCount	842
10.147.4.9	StreamBlockTransferSize	842
10.147.4.10	StreamBufferCountManual	842
10.147.4.11	StreamBufferCountMax	842
10.147.4.12	StreamBufferCountMode	842
10.147.4.13	StreamBufferCountResult	842
10.147.4.14	StreamBufferHandlingMode	843
10.147.4.15	StreamBufferUnderrunCount	843
10.147.4.16	StreamCRCCheckEnable	843
10.147.4.17	StreamDefaultBufferCount	843
10.147.4.18	StreamDefaultBufferCountMax	843

10.147.4.1	StreamDefaultBufferCountMode	843
10.147.4.2	StreamFailedBufferCount	843
10.147.4.3	StreamID	844
10.147.4.4	StreamTotalBufferCount	844
10.147.4.5	StreamType	844
10.148	U3V_CHUNK_TRAILER Struct Reference	844
10.148.1	Detailed Description	844
10.148.2	Member Data Documentation	844
10.148.2.1	ChunkID	844
10.148.2.2	ChunkLength	844
10.149	U3V_COMMAND_HEADER Struct Reference	845
10.149.1	Detailed Description	845
10.149.2	Member Data Documentation	845
10.149.2.1	CommandId	845
10.149.2.2	Flags	845
10.149.2.3	Length	845
10.149.2.4	Prefix	845
10.149.2.5	ReqId	845
10.150	U3V_EVENT_DATA Struct Reference	845
10.150.1	Detailed Description	846
10.150.2	Member Data Documentation	846
10.150.2.1	EventId	846
10.150.2.2	Reserved	846
10.150.2.3	Timestamp	846
10.151	U3V_EVENT_MESSAGE Struct Reference	846
10.151.1	Detailed Description	846
10.151.2	Member Data Documentation	847
10.151.2.1	CommandHeader	847
10.151.2.2	EventData	847
10.152	ValueNode Class Reference	847

10.152.1	Detailed Description	848
10.152.2	Constructor & Destructor Documentation	848
10.152.2.1	ValueNode()	848
10.152.2.2	ValueNode(std::shared_ptr< Node::NodeImpl > pValue)	848
10.152.2.3	~ValueNode()	848
10.152.3	Member Function Documentation	848
10.152.3.1	FromString(const GenICam::gcstring &ValueStr, bool Verify=true)	848
10.152.3.2	GetNode()	849
10.152.3.3	IsValueCacheValid() const	849
10.152.3.4	SetReference(INode *pBase)	849
10.152.3.5	ToString(bool Verify=false, bool IgnoreCache=false)	849
10.153	Version_t Struct Reference	849
10.153.1	Detailed Description	850
10.153.2	Member Data Documentation	850
10.153.2.1	Major	850
10.153.2.2	Minor	850
10.153.2.3	SubMinor	850
<b>11</b>	<b>File Documentation</b>	<b>851</b>
11.1	doc/Doxygen/spindocs/Licensing.dox File Reference	851
11.2	doc/Doxygen/spindocs/MainPage.dox File Reference	851
11.3	include/ArrivalEvent.h File Reference	851
11.4	include/AVIRecorder.h File Reference	853
11.5	include/BasePtr.h File Reference	853
11.6	include/Camera.h File Reference	855
11.7	include/CameraBase.h File Reference	857
11.8	include/CameraDefs.h File Reference	859
11.9	include/CameraList.h File Reference	889
11.10	include/CameraPtr.h File Reference	891
11.11	include/ChunkData.h File Reference	893
11.12	include/DeviceEvent.h File Reference	895

11.13include/Event.h File Reference . . . . .	897
11.14include/Exception.h File Reference . . . . .	899
11.15include/Image.h File Reference . . . . .	900
11.16include/ImageEvent.h File Reference . . . . .	902
11.17include/ImagePtr.h File Reference . . . . .	903
11.18include/ImageStatistics.h File Reference . . . . .	905
11.19include/Interface.h File Reference . . . . .	906
11.20include/Interface/IArrivalEvent.h File Reference . . . . .	908
11.21include/Interface/ICameraBase.h File Reference . . . . .	910
11.22include/Interface/ICameraList.h File Reference . . . . .	912
11.23include/Interface/IChunkData.h File Reference . . . . .	914
11.24include/Interface/IDeviceEvent.h File Reference . . . . .	916
11.25include/Interface/IImage.h File Reference . . . . .	918
11.26include/Interface/IImageEvent.h File Reference . . . . .	920
11.27include/Interface/IImageStatistics.h File Reference . . . . .	922
11.28include/Interface/IInterface.h File Reference . . . . .	924
11.29include/Interface/IInterfaceEvent.h File Reference . . . . .	926
11.30include/Interface/IInterfaceList.h File Reference . . . . .	928
11.31include/Interface/ILoggingEvent.h File Reference . . . . .	929
11.32include/Interface/IRemovalEvent.h File Reference . . . . .	931
11.33include/Interface/ISystem.h File Reference . . . . .	933
11.34include/InterfaceEvent.h File Reference . . . . .	934
11.35include/InterfaceList.h File Reference . . . . .	936
11.36include/InterfacePtr.h File Reference . . . . .	937
11.37include/LoggingEvent.h File Reference . . . . .	939
11.38include/LoggingEventData.h File Reference . . . . .	940
11.39include/LoggingEventDataPtr.h File Reference . . . . .	942
11.40include/RemovalEvent.h File Reference . . . . .	944
11.41include/SpinGenApi/Autovector.h File Reference . . . . .	946
11.42include/SpinGenApi/Base.h File Reference . . . . .	947

11.43include/SpinGenApi/BooleanNode.h File Reference . . . . .	948
11.44include/SpinGenApi/CategoryNode.h File Reference . . . . .	950
11.45include/SpinGenApi/ChunkAdapter.h File Reference . . . . .	952
11.46include/SpinGenApi/ChunkAdapterDcam.h File Reference . . . . .	954
11.47include/SpinGenApi/ChunkAdapterGeneric.h File Reference . . . . .	956
11.48include/SpinGenApi/ChunkAdapterGEV.h File Reference . . . . .	958
11.49include/SpinGenApi/ChunkAdapterU3V.h File Reference . . . . .	960
11.50include/SpinGenApi/ChunkPort.h File Reference . . . . .	962
11.51include/SpinGenApi/CommandNode.h File Reference . . . . .	964
11.52include/SpinGenApi/Compatibility.h File Reference . . . . .	967
11.52.1 Macro Definition Documentation . . . . .	968
11.52.1.1 FMT_I64 . . . . .	968
11.53include/SpinGenApi/Container.h File Reference . . . . .	968
11.54include/SpinGenApi/Counter.h File Reference . . . . .	968
11.55include/SpinGenApi/EnumClasses.h File Reference . . . . .	969
11.56include/SpinGenApi/EnumEntryNode.h File Reference . . . . .	971
11.57include/SpinGenApi/EnumNode.h File Reference . . . . .	973
11.58include/SpinGenApi/EnumNodeT.h File Reference . . . . .	975
11.59include/SpinGenApi/EventAdapter.h File Reference . . . . .	977
11.60include/SpinGenApi/EventAdapter1394.h File Reference . . . . .	979
11.61include/SpinGenApi/EventAdapterGeneric.h File Reference . . . . .	981
11.62include/SpinGenApi/EventAdapterGEV.h File Reference . . . . .	983
11.63include/SpinGenApi/EventAdapterU3V.h File Reference . . . . .	985
11.64include/SpinGenApi/EventPort.h File Reference . . . . .	987
11.65include/SpinGenApi/Filestream.h File Reference . . . . .	989
11.66include/SpinGenApi/FloatNode.h File Reference . . . . .	991
11.67include/SpinGenApi/FloatRegNode.h File Reference . . . . .	993
11.68include/SpinGenApi/GCBase.h File Reference . . . . .	995
11.69include/SpinGenApi/GCString.h File Reference . . . . .	996
11.69.1 Macro Definition Documentation . . . . .	997



11.69.1.1 GCSTRING_NPOS . . . . .	997
11.69.2 Function Documentation . . . . .	997
11.69.2.1 operator<<(std::ostream &ostr, const Spinnaker::GenICam::gcstring &str) . . . . .	997
11.69.2.2 operator>>(std::istream &istr, Spinnaker::GenICam::gcstring &str) . . . . .	997
11.70include/SpinGenApi/GCStringVector.h File Reference . . . . .	998
11.71include/SpinGenApi/GCSynch.h File Reference . . . . .	998
11.72include/SpinGenApi/GCTypes.h File Reference . . . . .	999
11.72.1 Macro Definition Documentation . . . . .	1000
11.72.1.1 __STDC_CONSTANT_MACROS . . . . .	1000
11.72.1.2 __STDC_LIMIT_MACROS . . . . .	1000
11.72.1.3 GC_INT32_MAX . . . . .	1000
11.72.1.4 GC_INT32_MIN . . . . .	1000
11.72.1.5 GC_INT64_MAX . . . . .	1000
11.72.1.6 GC_INT64_MIN . . . . .	1000
11.72.1.7 GC_INT8_MAX . . . . .	1000
11.72.1.8 GC_INT8_MIN . . . . .	1000
11.72.1.9 GC_UINT32_MAX . . . . .	1000
11.72.1.10GC_UINT64_MAX . . . . .	1001
11.72.1.11GC_UINT8_MAX . . . . .	1001
11.73include/SpinGenApi/GCUtilities.h File Reference . . . . .	1001
11.73.1 Macro Definition Documentation . . . . .	1004
11.73.1.1 __ERR__ . . . . .	1004
11.73.1.2 __LINE_STR__ . . . . .	1004
11.73.1.3 __LOCATION__ . . . . .	1004
11.73.1.4 __OUTPUT_FORMATER__ . . . . .	1004
11.73.1.5 __TODO__ . . . . .	1004
11.73.1.6 __WARN__ . . . . .	1004
11.73.1.7 _TO_STRING . . . . .	1004
11.73.1.8 EXPAND_TO_STRINGISE . . . . .	1004
11.73.1.9 GC_COUNTOF . . . . .	1004

11.73.1.10GENICAM_DEPRECATED . . . . .	1004
11.73.1.11GENICAM_UNUSED . . . . .	1004
11.73.1.12USE_TEMP_CACHE_FILE . . . . .	1004
11.73.1.13USE_TEMP_CACHE_FILE . . . . .	1004
11.74include/SpinGenApi/IBoolean.h File Reference . . . . .	1005
11.75include/SpinGenApi/ICategory.h File Reference . . . . .	1007
11.76include/SpinGenApi/IChunkPort.h File Reference . . . . .	1009
11.77include/SpinGenApi/ICommand.h File Reference . . . . .	1011
11.78include/SpinGenApi/IDestroy.h File Reference . . . . .	1013
11.79include/SpinGenApi/IDeviceInfo.h File Reference . . . . .	1015
11.80include/SpinGenApi/IEnumEntry.h File Reference . . . . .	1017
11.81include/SpinGenApi/IEnumeration.h File Reference . . . . .	1019
11.82include/SpinGenApi/IEnumerationT.h File Reference . . . . .	1021
11.83include/SpinGenApi/IFloat.h File Reference . . . . .	1023
11.84include/SpinGenApi/IInteger.h File Reference . . . . .	1025
11.85include/SpinGenApi/INode.h File Reference . . . . .	1027
11.86include/SpinGenApi/INodeMap.h File Reference . . . . .	1030
11.87include/SpinGenApi/INodeMapDyn.h File Reference . . . . .	1032
11.88include/SpinGenApi/IntegerNode.h File Reference . . . . .	1034
11.89include/SpinGenApi/IntRegNode.h File Reference . . . . .	1036
11.90include/SpinGenApi/IPort.h File Reference . . . . .	1038
11.91include/SpinGenApi/IPortConstruct.h File Reference . . . . .	1039
11.92include/SpinGenApi/IPortRecorder.h File Reference . . . . .	1041
11.93include/SpinGenApi/IRegister.h File Reference . . . . .	1043
11.94include/SpinGenApi/ISelector.h File Reference . . . . .	1045
11.95include/SpinGenApi/ISelectorDigit.h File Reference . . . . .	1046
11.96include/SpinGenApi/IString.h File Reference . . . . .	1048
11.97include/SpinGenApi/IValue.h File Reference . . . . .	1050
11.98include/SpinGenApi/Node.h File Reference . . . . .	1051
11.99include/SpinGenApi/NodeCallback.h File Reference . . . . .	1053

11.100	<a href="#">include/SpinGenApi/NodeCallbackImpl.h File Reference</a>	1055
11.101	<a href="#">include/SpinGenApi/NodeMap.h File Reference</a>	1056
11.102	<a href="#">include/SpinGenApi/NodeMapFactory.h File Reference</a>	1058
11.103	<a href="#">include/SpinGenApi/NodeMapRef.h File Reference</a>	1059
11.104	<a href="#">include/SpinGenApi/Persistence.h File Reference</a>	1060
11.105	<a href="#">include/SpinGenApi/Pointer.h File Reference</a>	1062
11.106	<a href="#">include/SpinGenApi/PortImpl.h File Reference</a>	1065
11.107	<a href="#">include/SpinGenApi/PortNode.h File Reference</a>	1066
11.108	<a href="#">include/SpinGenApi/PortRecorder.h File Reference</a>	1068
11.109	<a href="#">include/SpinGenApi/PortReplay.h File Reference</a>	1069
11.110	<a href="#">include/SpinGenApi/PortWriteList.h File Reference</a>	1070
11.111	<a href="#">include/SpinGenApi/Reference.h File Reference</a>	1072
11.112	<a href="#">include/SpinGenApi/RegisterNode.h File Reference</a>	1073
11.113	<a href="#">include/SpinGenApi/RegisterPortImpl.h File Reference</a>	1075
11.114	<a href="#">include/SpinGenApi/SelectorSet.h File Reference</a>	1075
11.115	<a href="#">include/SpinGenApi/SpinnakerGenApi.h File Reference</a>	1076
11.116	<a href="#">include/SpinGenApi/SpinTestCamera.h File Reference</a>	1078
11.117	<a href="#">include/SpinGenApi/StringNode.h File Reference</a>	1078
11.118	<a href="#">include/SpinGenApi/StringRegNode.h File Reference</a>	1080
11.119	<a href="#">include/SpinGenApi/StructPort.h File Reference</a>	1082
11.120	<a href="#">include/SpinGenApi/Synch.h File Reference</a>	1082
11.121	<a href="#">include/SpinGenApi/Types.h File Reference</a>	1083
11.121.1	<a href="#">Macro Definition Documentation</a>	1086
11.121.1.1	<a href="#">interface</a>	1086
11.122	<a href="#">include/SpinGenApi/ValueNode.h File Reference</a>	1087
11.123	<a href="#">include/Spinnaker.h File Reference</a>	1089
11.124	<a href="#">include/SpinnakerDefs.h File Reference</a>	1090
11.125	<a href="#">include/SpinnakerPlatform.h File Reference</a>	1095
11.126	<a href="#">include/SpinUpdate.h File Reference</a>	1095
11.126.1	<a href="#">Macro Definition Documentation</a>	1096

11.126.1.1SPINUPDATE_API . . . . .	1096
11.126.2Function Documentation . . . . .	1096
11.126.2.1GetErrorMessage() . . . . .	1096
11.126.2.2SetMessageCallback(UpdatorMessageCallback messageCallbackFunction) . . . . .	1096
11.126.2.3SetProgressCallback(UpdatorProgressCallback progressCallbackFunction) . . . . .	1096
11.126.2.4UpdateFirmware(const char *args) . . . . .	1096
11.126.2.5UpdateFirmwareConsole(int argc, char **argv) . . . . .	1096
11.126.3Variable Documentation . . . . .	1096
11.126.3.1UpdatorMessageCallback . . . . .	1096
11.126.3.2UpdatorProgressCallback . . . . .	1097
11.127include/SpinVideo.h File Reference . . . . .	1097
11.128include/SpinVideoDefs.h File Reference . . . . .	1097
11.129include/System.h File Reference . . . . .	1098
11.129.1Macro Definition Documentation . . . . .	1100
11.129.1.1FLIR_SPINNAKER_VERSION_BUILD . . . . .	1100
11.129.1.2FLIR_SPINNAKER_VERSION_MAJOR . . . . .	1100
11.129.1.3FLIR_SPINNAKER_VERSION_MINOR . . . . .	1100
11.129.1.4FLIR_SPINNAKER_VERSION_TYPE . . . . .	1100
11.130include/SystemPtr.h File Reference . . . . .	1100
11.131include/TransportLayerDefs.h File Reference . . . . .	1102
11.132include/TransportLayerDevice.h File Reference . . . . .	1104
11.133include/TransportLayerInterface.h File Reference . . . . .	1106
11.134include/TransportLayerStream.h File Reference . . . . .	1108
<b>Index</b>	<b>1111</b>

# Chapter 1

## Introduction

The [Spinnaker](#) application programming interface (API) is used to interface with FLIR's USB3 Vision and GigE Vision cameras.



## Chapter 2

# Software Licensing Information

Table 2.1 License table

Component	License
<a href="#">Spinnaker</a>	Copyright © 2017 FLIR Integrated Imaging Solutions, Inc. All Rights Reserved. This software is the confidential and proprietary information of FLIR Integrated Imaging Solutions, Inc. ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with FLIR Integrated Imaging Solutions, Inc. (FLIR). FLIR MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FLIR SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.
GenICam	<a href="#">GenICam License</a>
AdapterList	<a href="#">The Code Project Open License (CPO-OL)</a>
Make ListView.ScrollIntoView Scroll the Item into the Center of the ListView	<a href="#">WP:CC-BY-SA License</a>
Work with Bitmaps Faster in C#	<a href="#">The Code Project Open License (CPO-OL) 1.02</a>
FreeImage	<a href="#">FreeImage public license</a>
Boost	<a href="#">Boost Software License</a>
Libusb	<a href="#">LGPLv2.1 License</a>
Libraw1394	<a href="#">LGPLv2.0 License</a>
FFMPEG	<a href="#">LGPLv2.1 License</a>
log4Net	<a href="#">Apache license 2.0</a>
log4Cpp	<a href="#">LGPL License</a>

The licenses mentioned above can also be found in the [Spinnaker](#) installed license folder.





## Chapter 3

# Module Index

### 3.1 Modules

Here is a list of all modules:

Spinnaker Event Classes . . . . .	27
ArrivalEvent Class . . . . .	29
DeviceEvent Class . . . . .	139
Event Class . . . . .	140
ImageEvent Class . . . . .	143
InterfaceEvent Class . . . . .	147
LoggingEvent Class . . . . .	150
Logging Event Class . . . . .	151
LoggingEventDataPtr Class . . . . .	152
RemovalEvent Class . . . . .	153
Spinnaker Classes . . . . .	30
AVI Recorder Class . . . . .	33
BasePtr Class . . . . .	35
Camera Class . . . . .	36
Camera Base Class . . . . .	37
CameraDefs Class . . . . .	38
Camera List Class . . . . .	136
CameraPtr Class . . . . .	137
ChunkData Class . . . . .	138
Exception Class . . . . .	141
Image Class . . . . .	142
ImagePtr Class . . . . .	144
ImageStatistics Class . . . . .	145
Interface Class . . . . .	146
InterfaceList Class . . . . .	148
InterfacePtr Class . . . . .	149
Spinnaker Video Class . . . . .	169
System Class . . . . .	171
SystemPtr Class . . . . .	172
Camera Base Interface Class . . . . .	183
IChunkData Class . . . . .	184
IImage Class . . . . .	185
IImageStatistics Class . . . . .	186
IInterface Class . . . . .	187
IInterfaceList Class . . . . .	188

ISystem Class . . . . .	189
Spinnaker Headers . . . . .	154
Spinnaker.h . . . . .	156
Spinnaker Definitions . . . . .	157
Spinnaker Platform . . . . .	168
Spinnaker Video Definitions . . . . .	170
Spinnaker QuickSpin Classes . . . . .	173
TransportLayerDefs Class . . . . .	174
TransportLayerDevice Class . . . . .	180
TransportLayerInterface Class . . . . .	181
TransportLayerStream Class . . . . .	182
Spinnaker GenApi Classes . . . . .	190
AutoVector Class . . . . .	199
BooleanNode Class . . . . .	203
CategoryNode Class . . . . .	204
ChunkAdapter Class . . . . .	205
ChunkAdapterDcam Class . . . . .	206
ChunkAdapterGeneric Class . . . . .	207
ChunkAdapterGEV Class . . . . .	208
ChunkPort Class . . . . .	209
CommandNode Class . . . . .	210
Container Class . . . . .	211
Counter Class . . . . .	212
EnumClasses Class . . . . .	213
EnumEntryNode Class . . . . .	215
EnumNode Class . . . . .	216
EnumNodeT Class . . . . .	217
EventAdapter Class . . . . .	218
EventAdapter1394 Class . . . . .	219
EventAdapterGeneric Class . . . . .	220
EventAdapterGEV Class . . . . .	221
EventAdapterU3V Class . . . . .	222
EventPort Class . . . . .	223
Filestream Class . . . . .	224
FloatNode Class . . . . .	225
FloatRegNode Class . . . . .	226
GCString Class . . . . .	227
GCSynch Class . . . . .	228
GCTypes Class . . . . .	229
IntegerNode Class . . . . .	266
IntRegNode Class . . . . .	267
IString Class . . . . .	277
IValue Class . . . . .	278
Node Class . . . . .	280
NodeCallback Class . . . . .	281
NodeMap Class . . . . .	283
NodeMapFactory Class . . . . .	284
NodeMapRef Class . . . . .	286
Persistence Class . . . . .	287
Pointer Class . . . . .	288
PortImpl Class . . . . .	293
PortNode Class . . . . .	294
PortRecorder Class . . . . .	295
PortReplay Class . . . . .	296
PortWriteList Class . . . . .	297
RegisterNode Class . . . . .	299
RegisterPortImpl Class . . . . .	300

SelectorSet Class . . . . .	301
SpinTestCamera Class . . . . .	302
StringNode Class . . . . .	303
StringRegNode Class . . . . .	304
StructPort Class . . . . .	305
Synch Class . . . . .	306
ValueNode Class . . . . .	316
ChunkAdapterU3V Class . . . . .	317
IPortRecorder Interface . . . . .	270
Spinnaker GenApi Interfaces . . . . .	200
IBase Interface . . . . .	202
IBoolean Interface . . . . .	235
ICategory Interfaces . . . . .	237
IChunkPort Interface . . . . .	238
ICommand Interface . . . . .	240
IDestroy Interface . . . . .	241
IDeviceInfo Interface . . . . .	242
IEnumEntry Interface . . . . .	244
IEnumeration Interface . . . . .	245
IEnumerationT Interface . . . . .	247
IFloat Interface . . . . .	249
IInteger Interface . . . . .	252
INode Interface . . . . .	253
INodeMap Interface . . . . .	260
INodeMapDyn Interface . . . . .	262
IPort Interface . . . . .	268
IPortConstruct Interface . . . . .	269
IPortRecorder Interface . . . . .	270
IRegister Interfaces . . . . .	272
ISelector Interface . . . . .	274
ISelectorDigit Interface . . . . .	275
Reference Interfaces . . . . .	298
Spinnaker GenApi Utilities . . . . .	230
GCUilities Utility . . . . .	231
Spinnaker GenApi Enums . . . . .	307
Types Enums . . . . .	308



## Chapter 4

# Namespace Index

### 4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Spinnaker</a> . . . . .	319
<a href="#">Spinnaker::GenApi</a> . . . . .	356
<a href="#">Spinnaker::GenICam</a> . . . . .	372
<a href="#">Spinnaker::Video</a> . . . . .	374



## Chapter 5

# Hierarchical Index

### 5.1 Class Hierarchy

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

ActionCommandResult . . . . .	375
AttachStatistics_t . . . . .	377
AutoLock . . . . .	378
AutoLock . . . . .	379
AVIOption . . . . .	379
BasePtr< T, B > . . . . .	380
BasePtr< Camera, ICameraBase > . . . . .	380
CameraPtr . . . . .	526
BasePtr< IImage > . . . . .	380
ImagePtr . . . . .	726
BasePtr< IInterface > . . . . .	380
InterfacePtr . . . . .	748
BasePtr< ISystem > . . . . .	380
SystemPtr . . . . .	822
BasePtr< LoggingEventData > . . . . .	380
LoggingEventDataPtr . . . . .	766
basic_istream	
IDevFileStreamBase< CharType, Traits > . . . . .	680
basic_ostream	
ODevFileStreamBase< CharType, Traits > . . . . .	785
basic_streambuf	
IDevFileStreamBuf< CharType, Traits > . . . . .	682
ODevFileStreamBuf< CharType, Traits > . . . . .	787
BMPOption . . . . .	382
CChunkAdapter . . . . .	529
CChunkAdapterDcam . . . . .	531
CChunkAdapterGeneric . . . . .	533
CChunkAdapterGEV . . . . .	535
CChunkAdapterU3V . . . . .	537
CDataStruct	
CTestPortStruct< CDataStruct > . . . . .	611
CEventAdapter . . . . .	546
CEventAdapter1394 . . . . .	548

CEventAdapterGeneric	549
CEventAdapterGEV	551
CEventAdapterU3V	553
CGeneric_XMLLoaderParams	562
CNodeMapRefT< GenApi::CGeneric_XMLLoaderParams >	591
CNodeMapRef	589
CGlobalLock	562
CGlobalLockUnlocker	564
CLock	575
CLockEx	578
CLock	577
CLockEx	579
CNodeCallback	580
Function_NodeCallback< Function >	652
Member_NodeCallback< Client, Member >	767
CNodeMapFactory	582
Counter	598
CPointer< T, B >	599
CPointer< IFloat, IBase >	599
CFloatPtr	560
CPointer< INode, IBase >	599
DCAM_CHECKSUM	614
DCAM_CHUNK_TRAILER	615
double_autovector_t	618
EAccessModeClass	619
ECachingModeClass	620
EDisplayNotationClass	620
EEndianessClass	621
EGenApiSchemaVersionClass	622
EInputDirectionClass	623
ENamespaceClass	624
ERepresentationClass	632
ESignClass	632
ESlopeClass	633
EStandardNameSpaceClass	634
Event	635
IArrivalEvent	666
ArrivalEvent	376
IInterfaceEvent	699
InterfaceEvent	743
IDeviceEvent	683
DeviceEvent	615
IImageEvent	692
ImageEvent	724
ILoggingEvent	702
LoggingEvent	761
IRemovalEvent	752
IInterfaceEvent	699
RemovalEvent	803
EVisibilityClass	637
exception	
Exception	638
EYesNoClass	641
FileProtocolAdapter	642
gcstring	654
GVCP_CHUNK_TRAILER	657



GVCP_EVENT_ITEM . . . . .	658
GVCP_EVENT_ITEM_BASIC . . . . .	659
GVCP_EVENT_ITEM_EXTENDED_ID . . . . .	659
GVCP_EVENT_REQUEST . . . . .	660
GVCP_EVENT_REQUEST_EXTENDED_ID . . . . .	661
GVCP_EVENTDATA_REQUEST . . . . .	662
GVCP_EVENTDATA_REQUEST_EXTENDED_ID . . . . .	663
GVCP_REQUEST_HEADER . . . . .	663
H264Option . . . . .	664
IBoolean	
BooleanNode . . . . .	383
ICameraBase . . . . .	667
CameraBase . . . . .	513
Camera . . . . .	386
ICameraList . . . . .	672
CameraList . . . . .	522
ICategory	
CategoryNode . . . . .	527
ICChunkData . . . . .	675
ChunkData . . . . .	566
IChunkPort	
PortNode . . . . .	790
PortReplay . . . . .	797
PortRecorder . . . . .	794
ICommand	
CommandNode . . . . .	595
IDeviceInfo	
NodeMap . . . . .	778
SpinTestCamera . . . . .	806
IEnumEntry	
EnumEntryNode . . . . .	625
IEnumeration	
EnumNode . . . . .	628
CEnumerationTRef< EnumT > . . . . .	542
IEnumerationT	
CEnumerationTRef< EnumT > . . . . .	542
IFloat	
FloatNode . . . . .	645
FloatRegNode . . . . .	650
IImage . . . . .	685
Image . . . . .	704
IImageStatistics . . . . .	693
ImageStatistics . . . . .	728
IInteger	
IntegerNode . . . . .	735
IntRegNode . . . . .	750
IInterface . . . . .	696
Interface . . . . .	739
IInterfaceList . . . . .	700
InterfaceList . . . . .	745
INode	
Node . . . . .	770
CSelectorSet . . . . .	609
PortNode . . . . .	790
ValueNode . . . . .	847

BooleanNode . . . . .	383
CategoryNode . . . . .	527
CommandNode . . . . .	595
EnumEntryNode . . . . .	625
EnumNode . . . . .	628
FloatNode . . . . .	645
IntegerNode . . . . .	735
RegisterNode . . . . .	800
FloatRegNode . . . . .	650
IntRegNode . . . . .	750
StringRegNode . . . . .	813
StringNode . . . . .	809
StringRegNode . . . . .	813
INodeMap	
NodeMap . . . . .	778
int64_autovector_t . . . . .	733
IPersistScript	
CFeatureBag . . . . .	558
IPortConstruct	
CChunkPort . . . . .	538
CEventPort . . . . .	554
CPortImpl . . . . .	602
CRegisterPortImpl . . . . .	607
CTestPortStruct< CDataStruct > . . . . .	611
PortNode . . . . .	790
IPortRecorder	
PortNode . . . . .	790
PortRecorder . . . . .	794
IPortReplay	
CPortImpl . . . . .	602
PortReplay . . . . .	797
IPortWriteList	
CPortWriteList . . . . .	605
IRegister	
RegisterNode . . . . .	800
IString	
StringNode . . . . .	809
ISystem . . . . .	753
System . . . . .	815
IValue	
ValueNode . . . . .	847
JPEGOption . . . . .	756
JPG2Option . . . . .	757
LibraryVersion . . . . .	758
LockableObject< Object >::Lock . . . . .	759
LockableObject< Object > . . . . .	760
LoggingEventData . . . . .	763
MJPGOption . . . . .	769
CNodeMapFactory::NodeStatistics_t . . . . .	784
PGMOption . . . . .	788
PNGOption . . . . .	789
PPMOption . . . . .	799
SingleChunkData_t . . . . .	805
SingleChunkDataStr_t . . . . .	805
SpinVideo . . . . .	806
TIFFOption . . . . .	823
TransportLayerDevice . . . . .	825
TransportLayerInterface . . . . .	832

TransportLayerStream . . . . .	839
U3V_CHUNK_TRAILER . . . . .	844
U3V_COMMAND_HEADER . . . . .	845
U3V_EVENT_DATA . . . . .	845
U3V_EVENT_MESSAGE . . . . .	846
Version_t . . . . .	849
TCameraParams	
CNodeMapRefT< TCameraParams > . . . . .	591



## Chapter 6

# Class Index

### 6.1 Class List

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

<a href="#">ActionCommandResult</a>	
Action Command Result . . . . .	375
<a href="#">ArrivalEvent</a>	
An event handler for capturing the device arrival event . . . . .	376
<a href="#">AttachStatistics_t</a>	
Delivers information about the attached chunks and nodes . . . . .	377
<a href="#">AutoLock</a>	378
<a href="#">AutoLock</a>	379
<a href="#">AVIOption</a>	
Options for saving AVI files . . . . .	379
<a href="#">BasePtr&lt; T, B &gt;</a>	
The base class of the <a href="#">SystemPtr</a> , <a href="#">CameraPtr</a> , <a href="#">InterfacePtr</a> , <a href="#">ImagePtr</a> and <a href="#">LoggingEventDataPtr</a> objects . . . . .	380
<a href="#">BMPOption</a>	
Options for saving Bitmap image . . . . .	382
<a href="#">BooleanNode</a>	
<a href="#">Interface</a> for string properties . . . . .	383
<a href="#">Camera</a>	
The camera object class . . . . .	386
<a href="#">CameraBase</a>	
The base class for the camera object . . . . .	513
<a href="#">CameraList</a>	
Used to hold a list of camera objects . . . . .	522
<a href="#">CameraPtr</a>	
A reference tracked pointer to a camera object . . . . .	526
<a href="#">CategoryNode</a>	
<a href="#">Interface</a> for string properties . . . . .	527
<a href="#">CChunkAdapter</a>	
Connects a chunked buffer to a node map . . . . .	529
<a href="#">CChunkAdapterDcam</a>	
Connects a chunked DCAM buffer to a node map . . . . .	531
<a href="#">CChunkAdapterGeneric</a>	533
<a href="#">CChunkAdapterGEV</a>	
Connects a chunked DCAM buffer to a node map . . . . .	535
<a href="#">CChunkAdapterU3V</a>	
Connects a chunked U3V buffer to a node map . . . . .	537

<a href="#">CChunkPort</a>	
Port attachable to a chunk in a buffer . . . . .	538
<a href="#">CEnumerationTRef&lt; EnumT &gt;</a>	
Interface for string properties . . . . .	542
<a href="#">CEventAdapter</a>	
Delivers Events to ports . . . . .	546
<a href="#">CEventAdapter1394</a>	
Distribute the events to the node map . . . . .	548
<a href="#">CEventAdapterGeneric</a>	
Connects a generic event to a node map . . . . .	549
<a href="#">CEventAdapterGEV</a>	
Connects a GigE <a href="#">Event</a> to a node map . . . . .	551
<a href="#">CEventAdapterU3V</a>	
Connects a U3V <a href="#">Event</a> to a node map . . . . .	553
<a href="#">CEventPort</a>	
Port attachable to an event . . . . .	554
<a href="#">CFeatureBag</a>	
Bag holding streamable features of a nodetree . . . . .	558
<a href="#">CFloatPtr</a>	
SmartPointer for IFloat interface pointer . . . . .	560
<a href="#">CGeneric_XMLLoaderParams</a>	
Empty base class used by class <a href="#">CNodeMapRef</a> as generic template argument . . . . .	562
<a href="#">CGlobalLock</a>	
Named global lock which can be used over process boundaries . . . . .	562
<a href="#">CGlobalLockUnlocker</a>	
Unlocks the global lock object on destruction . . . . .	564
<a href="#">ChunkData</a>	
The chunk data which contains additional information about an image . . . . .	566
<a href="#">CLock</a>	
A lock class . . . . .	575
<a href="#">CLock</a>	
A lock class . . . . .	577
<a href="#">CLockEx</a>	
This class is for testing purposes only . . . . .	578
<a href="#">CLockEx</a>	
This class is for testing purposes only . . . . .	579
<a href="#">CNodeCallback</a>	
Callback body instance for INode pointers . . . . .	580
<a href="#">CNodeMapFactory</a>	
The node map factory is used for creating node maps from camera description files . . . . .	582
<a href="#">CNodeMapRef</a>	
SmartPointer for NodeMaps with create function . . . . .	589
<a href="#">CNodeMapRefT&lt; TCameraParams &gt;</a>	
SmartPointer template for NodeMaps with create function . . . . .	591
<a href="#">CommandNode</a>	
Interface for string properties . . . . .	595
<a href="#">Counter</a>	
Definition of a simple <a href="#">Counter</a> class . . . . .	598
<a href="#">CPointer&lt; T, B &gt;</a>	
Encapsulates a <a href="#">GenApi</a> pointer dealing with the dynamic_cast automatically . . . . .	599
<a href="#">CPortImpl</a>	
Standard implementation for a port . . . . .	602
<a href="#">CPortWriteList</a>	
Container holding a list of port write commands . . . . .	605
<a href="#">CRegisterPortImpl</a>	
Standard implementation for a port using a register based transport layer . . . . .	607
<a href="#">CSelectorSet</a>	
The set of selectors selecting a given node . . . . .	609

<a href="#">CTestPortStruct&lt; CDataStruct &gt;</a>	
Implements a register spaces based on a C++ struct	611
<a href="#">DCAM_CHECKSUM</a>	614
<a href="#">DCAM_CHUNK_TRAILER</a>	615
<a href="#">DeviceEvent</a>	
A handler to device events	615
<a href="#">double_autovector_t</a>	
Vector of doubles with reference counting	618
<a href="#">EAccessModeClass</a>	
Holds conversion methods for the access mode enumeration	619
<a href="#">ECachingModeClass</a>	
Holds conversion methods for the caching mode enumeration	620
<a href="#">EDisplayNotationClass</a>	
Holds conversion methods for the notation type of floats	620
<a href="#">EEndianessClass</a>	
Holds conversion methods for the endianess enumeration	621
<a href="#">EGenApiSchemaVersionClass</a>	
Helper class converting EGenApiSchemaVersion from and to string	622
<a href="#">EInputDirectionClass</a>	
Holds conversion methods for the notation type of floats	623
<a href="#">ENameSpaceClass</a>	
Holds conversion methods for the namespace enumeration	624
<a href="#">EnumEntryNode</a>	
Interface for string properties	625
<a href="#">EnumNode</a>	
Interface for string properties	628
<a href="#">ERepresentationClass</a>	
Holds conversion methods for the representation enumeration	632
<a href="#">ESignClass</a>	
Holds conversion methods for the sign enumeration	632
<a href="#">ESlopeClass</a>	
Holds conversion methods for the converter formulas	633
<a href="#">EStandardNameSpaceClass</a>	
Holds conversion methods for the standard namespace enumeration	634
<a href="#">Event</a>	
The base class for all event types	635
<a href="#">EVisibilityClass</a>	
Holds conversion methods for the visibility enumeration	637
<a href="#">Exception</a>	
The <a href="#">Exception</a> object represents an error that is returned from the library	638
<a href="#">EYesNoClass</a>	
Holds conversion methods for the standard namespace enumeration	641
<a href="#">FileProtocolAdapter</a>	
Adapter between the std::iostreambuf and the SFNC Features representing the device file system	642
<a href="#">FloatNode</a>	
Interface for string properties	645
<a href="#">FloatRegNode</a>	
Interface for string properties	650
<a href="#">Function_NodeCallback&lt; Function &gt;</a>	
Container for a function pointer	652
<a href="#">gcstring</a>	654
<a href="#">GVCP_CHUNK_TRAILER</a>	
Header of a GVCP request packet	657
<a href="#">GVCP_EVENT_ITEM</a>	
Layout of a GVCP event item (Extended ID flag not set)	658
<a href="#">GVCP_EVENT_ITEM_BASIC</a>	
Layout of a GVCP event item (common to all types)	659

<a href="#">GVCP_EVENT_ITEM_EXTENDED_ID</a>	
Layout of a GVCP event item (Extended ID flag set)	659
<a href="#">GVCP_EVENT_REQUEST</a>	
Layout of a GVCP event request packet (Extended ID flag not set)	660
<a href="#">GVCP_EVENT_REQUEST_EXTENDED_ID</a>	
Layout of a GVCP event request packet (Extended ID flag set)	661
<a href="#">GVCP_EVENTDATA_REQUEST</a>	
Layout of a GVCP event data request packet (Extended ID flag not set)	662
<a href="#">GVCP_EVENTDATA_REQUEST_EXTENDED_ID</a>	
Layout of a GVCP event data request packet (Extended ID flag set)	663
<a href="#">GVCP_REQUEST_HEADER</a>	
Header of a GVCP request packet	663
<a href="#">H264Option</a>	
Options for saving H264 files	664
<a href="#">IArrivalEvent</a>	666
<a href="#">ICameraBase</a>	
The interface file for base class for the camera object	667
<a href="#">ICameraList</a>	
Used to hold a list of camera objects	672
<a href="#">IChunkData</a>	
The <a href="#">Interface</a> file for <a href="#">ChunkData</a>	675
<a href="#">IDevFileStreamBase&lt; CharType, Traits &gt;</a>	680
<a href="#">IDevFileStreamBuf&lt; CharType, Traits &gt;</a>	682
<a href="#">IDeviceEvent</a>	683
<a href="#">IImage</a>	
The interface file for <a href="#">Image</a>	685
<a href="#">IImageEvent</a>	692
<a href="#">IImageStatistics</a>	
The interface file for image statistics	693
<a href="#">IInterface</a>	
The interface file for <a href="#">Interface</a>	696
<a href="#">IInterfaceEvent</a>	699
<a href="#">IInterfaceList</a>	
The interface file for <a href="#">InterfaceList</a> class	700
<a href="#">ILoggingEvent</a>	702
<a href="#">Image</a>	
The image object class	704
<a href="#">ImageEvent</a>	
A handler for capturing image arrival events	724
<a href="#">ImagePtr</a>	
A reference tracked pointer to an image object	726
<a href="#">ImageStatistics</a>	
Represents image statistics for an image	728
<a href="#">int64_autovector_t</a>	
Vector of integers with reference counting	733
<a href="#">IntegerNode</a>	
<a href="#">Interface</a> for string properties	735
<a href="#">Interface</a>	
An interface object which holds a list of cameras	739
<a href="#">InterfaceEvent</a>	
A handler to device arrival and removal events on all interfaces	743
<a href="#">InterfaceList</a>	
A list of the available interfaces on the system	745
<a href="#">InterfacePtr</a>	
A reference tracked pointer to the interface object	748
<a href="#">IntRegNode</a>	
<a href="#">Interface</a> for string properties	750
<a href="#">IRemovalEvent</a>	752



<a href="#">ISystem</a>	
The interface file for <a href="#">System</a>	753
<a href="#">JPEGOption</a>	
Options for saving JPEG image	756
<a href="#">JPG2Option</a>	
Options for saving JPEG2000 image	757
<a href="#">LibraryVersion</a>	
Provides easier access to the current version of <a href="#">Spinnaker</a>	758
<a href="#">LockableObject&lt; Object &gt;::Lock</a>	
A scopelevel <a href="#">Lock</a> class	759
<a href="#">LockableObject&lt; Object &gt;</a>	
Instance-Lock for an object	760
<a href="#">LoggingEvent</a>	
An event handler for capturing the device logging event	761
<a href="#">LoggingEventData</a>	
The <a href="#">LoggingEventData</a> object	763
<a href="#">LoggingEventDataPtr</a>	
A reference tracked pointer to the <a href="#">LoggingEvent</a> object	766
<a href="#">Member_NodeCallback&lt; Client, Member &gt;</a>	
Container for a member function pointer	767
<a href="#">MJPGOption</a>	
Options for saving MJPG files	769
<a href="#">Node</a>	
Class common to all nodes	770
<a href="#">NodeMap</a>	
Smart pointer template for NodeMaps with create function	778
<a href="#">CNodeMapFactory::NodeStatistics_t</a>	784
<a href="#">ODevFileStreamBase&lt; CharType, Traits &gt;</a>	785
<a href="#">ODevFileStreamBuf&lt; CharType, Traits &gt;</a>	787
<a href="#">PGMOption</a>	
Options for saving PGM images	788
<a href="#">PNGOption</a>	
Options for saving PNG images	789
<a href="#">PortNode</a>	
Interface for value properties	790
<a href="#">PortRecorder</a>	
Interface for recording write commands on a port	794
<a href="#">PortReplay</a>	
Interface for replaying write commands on a port	797
<a href="#">PPMOption</a>	
Options for saving PPM images	799
<a href="#">RegisterNode</a>	
Interface for string properties	800
<a href="#">RemovalEvent</a>	
An event handler for capturing the device removal event	803
<a href="#">SingleChunkData_t</a>	805
<a href="#">SingleChunkDataStr_t</a>	805
<a href="#">SpinTestCamera</a>	806
<a href="#">SpinVideo</a>	
Provides the functionality for the user to record images to an AVI/MP4 file	806
<a href="#">StringNode</a>	
Interface for string properties	809
<a href="#">StringRegNode</a>	
Interface for string properties	813
<a href="#">System</a>	
The system object is used to retrieve the list of interfaces and cameras available	815
<a href="#">SystemPtr</a>	
A reference tracked pointer to a system object	822

<a href="#">TIFFOption</a>	
Options for saving TIFF images	823
<a href="#">TransportLayerDevice</a>	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	825
<a href="#">TransportLayerInterface</a>	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	832
<a href="#">TransportLayerStream</a>	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	839
<a href="#">U3V_CHUNK_TRAILER</a>	
Header of a GVCP request packet	844
<a href="#">U3V_COMMAND_HEADER</a>	
U3V/GenCP command header	845
<a href="#">U3V_EVENT_DATA</a>	
U3V/GenCP EVENT_CMD specific command data	845
<a href="#">U3V_EVENT_MESSAGE</a>	
Entire event data message (without the variable-sized data field)	846
<a href="#">ValueNode</a>	
Interface for value properties	847
<a href="#">Version_t</a>	
Version	849

## Chapter 7

# File Index

### 7.1 File List

Here is a list of all files with brief descriptions:

include/ArrivalEvent.h	851
include/AVIRecorder.h	853
include/BasePtr.h	853
include/Camera.h	855
include/CameraBase.h	857
include/CameraDefs.h	859
include/CameraList.h	889
include/CameraPtr.h	891
include/ChunkData.h	893
include/DeviceEvent.h	895
include/Event.h	897
include/Exception.h	899
include/Image.h	900
include/ImageEvent.h	902
include/ImagePtr.h	903
include/ImageStatistics.h	905
include/Interface.h	906
include/InterfaceEvent.h	934
include/InterfaceList.h	936
include/InterfacePtr.h	937
include/LoggingEvent.h	939
include/LoggingEventData.h	940
include/LoggingEventDataPtr.h	942
include/RemovalEvent.h	944
include/Spinnaker.h	1089
include/SpinnakerDefs.h	1090
include/SpinnakerPlatform.h	1095
include/SpinUpdate.h	1095
include/SpinVideo.h	1097
include/SpinVideoDefs.h	1097
include/System.h	1098
include/SystemPtr.h	1100
include/TransportLayerDefs.h	1102
include/TransportLayerDevice.h	1104
include/TransportLayerInterface.h	1106

include/TransportLayerStream.h	1108
include/Interface/IArrivalEvent.h	908
include/Interface/ICameraBase.h	910
include/Interface/ICameraList.h	912
include/Interface/IChunkData.h	914
include/Interface/IDeviceEvent.h	916
include/Interface/IImage.h	918
include/Interface/IImageEvent.h	920
include/Interface/IImageStatistics.h	922
include/Interface/IInterface.h	924
include/Interface/IInterfaceEvent.h	926
include/Interface/IInterfaceList.h	928
include/Interface/ILoggingEvent.h	929
include/Interface/IRemovalEvent.h	931
include/Interface/ISystem.h	933
include/SpinGenApi/Autovector.h	946
include/SpinGenApi/Base.h	947
include/SpinGenApi/BooleanNode.h	948
include/SpinGenApi/CategoryNode.h	950
include/SpinGenApi/ChunkAdapter.h	952
include/SpinGenApi/ChunkAdapterDcam.h	954
include/SpinGenApi/ChunkAdapterGeneric.h	956
include/SpinGenApi/ChunkAdapterGEV.h	958
include/SpinGenApi/ChunkAdapterU3V.h	960
include/SpinGenApi/ChunkPort.h	962
include/SpinGenApi/CommandNode.h	964
include/SpinGenApi/Compatibility.h	967
include/SpinGenApi/Container.h	968
include/SpinGenApi/Counter.h	968
include/SpinGenApi/EnumClasses.h	969
include/SpinGenApi/EnumEntryNode.h	971
include/SpinGenApi/EnumNode.h	973
include/SpinGenApi/EnumNodeT.h	975
include/SpinGenApi/EventAdapter.h	977
include/SpinGenApi/EventAdapter1394.h	979
include/SpinGenApi/EventAdapterGeneric.h	981
include/SpinGenApi/EventAdapterGEV.h	983
include/SpinGenApi/EventAdapterU3V.h	985
include/SpinGenApi/EventPort.h	987
include/SpinGenApi/Filestream.h	989
include/SpinGenApi/FloatNode.h	991
include/SpinGenApi/FloatRegNode.h	993
include/SpinGenApi/GCBase.h	995
include/SpinGenApi/GCString.h	996
include/SpinGenApi/GCStringVector.h	998
include/SpinGenApi/GCSynch.h	998
include/SpinGenApi/GCTypes.h	999
include/SpinGenApi/GCUtilities.h	1001
include/SpinGenApi/IBoolean.h	1005
include/SpinGenApi/ICategory.h	1007
include/SpinGenApi/IChunkPort.h	1009
include/SpinGenApi/ICommand.h	1011
include/SpinGenApi/IDestroy.h	1013
include/SpinGenApi/IDeviceInfo.h	1015
include/SpinGenApi/IEnumEntry.h	1017
include/SpinGenApi/IEnumeration.h	1019
include/SpinGenApi/IEnumerationT.h	1021
include/SpinGenApi/IFloat.h	1023

include/SpinGenApi/IInteger.h	1025
include/SpinGenApi/INode.h	1027
include/SpinGenApi/INodeMap.h	1030
include/SpinGenApi/INodeMapDyn.h	1032
include/SpinGenApi/IntegerNode.h	1034
include/SpinGenApi/IntRegNode.h	1036
include/SpinGenApi/IPort.h	1038
include/SpinGenApi/IPortConstruct.h	1039
include/SpinGenApi/IPortRecorder.h	1041
include/SpinGenApi/IRegister.h	1043
include/SpinGenApi/ISelector.h	1045
include/SpinGenApi/ISelectorDigit.h	1046
include/SpinGenApi/IString.h	1048
include/SpinGenApi/IValue.h	1050
include/SpinGenApi/Node.h	1051
include/SpinGenApi/NodeCallback.h	1053
include/SpinGenApi/NodeCallbackImpl.h	1055
include/SpinGenApi/NodeMap.h	1056
include/SpinGenApi/NodeMapFactory.h	1058
include/SpinGenApi/NodeMapRef.h	1059
include/SpinGenApi/Persistence.h	1060
include/SpinGenApi/Pointer.h	1062
include/SpinGenApi/PortImpl.h	1065
include/SpinGenApi/PortNode.h	1066
include/SpinGenApi/PortRecorder.h	1068
include/SpinGenApi/PortReplay.h	1069
include/SpinGenApi/PortWriteList.h	1070
include/SpinGenApi/Reference.h	1072
include/SpinGenApi/RegisterNode.h	1073
include/SpinGenApi/RegisterPortImpl.h	1075
include/SpinGenApi/SelectorSet.h	1075
include/SpinGenApi/SpinnakerGenApi.h	1076
include/SpinGenApi/SpinTestCamera.h	1078
include/SpinGenApi/StringNode.h	1078
include/SpinGenApi/StringRegNode.h	1080
include/SpinGenApi/StructPort.h	1082
include/SpinGenApi/Synch.h	1082
include/SpinGenApi/Types.h	1083
include/SpinGenApi/ValueNode.h	1087

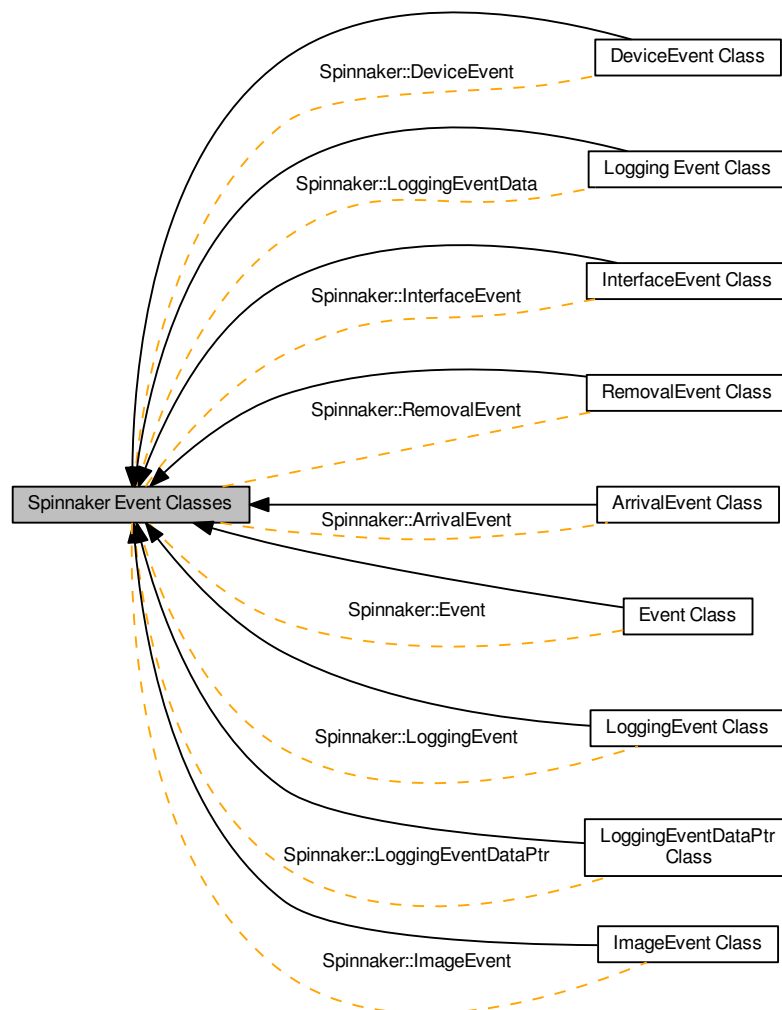


## Chapter 8

# Module Documentation

### 8.1 Spinnaker Event Classes

Collaboration diagram for Spinnaker Event Classes:



## Modules

- [ArrivalEvent Class](#)
- [DeviceEvent Class](#)
- [Event Class](#)
- [ImageEvent Class](#)
- [InterfaceEvent Class](#)
- [LoggingEvent Class](#)
- [Logging Event Class](#)
- [LoggingEventDataPtr Class](#)
- [RemovalEvent Class](#)

## Classes

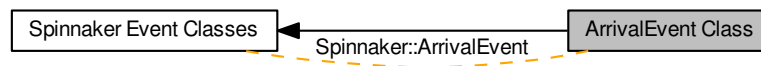
- class [ArrivalEvent](#)  
*An event handler for capturing the device arrival event.*
- class [DeviceEvent](#)  
*A handler to device events.*
- class [Event](#)  
*The base class for all event types.*
- class [ImageEvent](#)  
*A handler for capturing image arrival events.*
- class [InterfaceEvent](#)  
*A handler to device arrival and removal events on all interfaces.*
- class [LoggingEvent](#)  
*An event handler for capturing the device logging event.*
- class [LoggingEventData](#)  
*The [LoggingEventData](#) object.*
- class [LoggingEventDataPtr](#)  
*A reference tracked pointer to the [LoggingEvent](#) object.*
- class [RemovalEvent](#)  
*An event handler for capturing the device removal event.*

### 8.1.1 Detailed Description



## 8.2 ArrivalEvent Class

Collaboration diagram for ArrivalEvent Class:



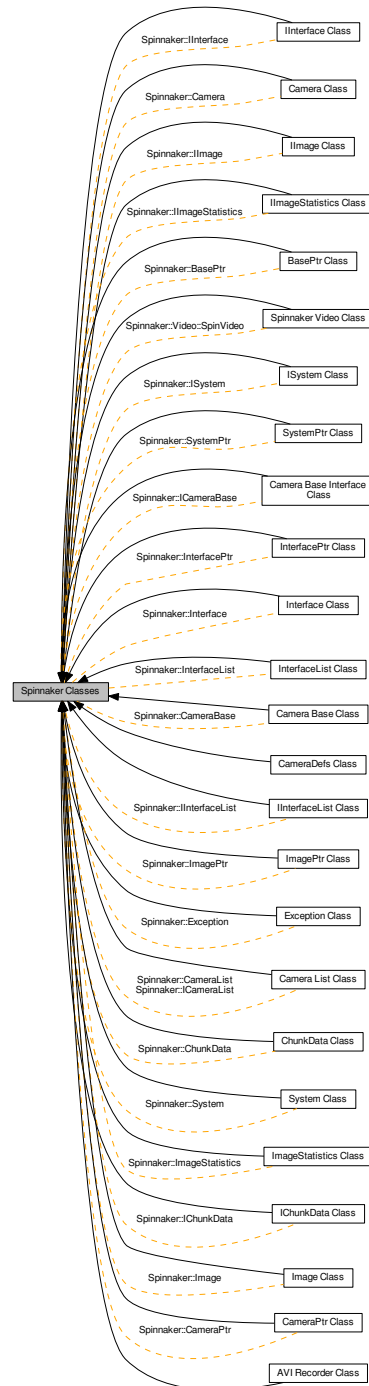
### Classes

- class [ArrivalEvent](#)  
*An event handler for capturing the device arrival event.*

### 8.2.1 Detailed Description

## 8.3 Spinnaker Classes

Collaboration diagram for Spinnaker Classes:



### Modules

- [AVI Recorder Class](#)
- [BasePtr Class](#)

- [Camera Class](#)
- [Camera Base Class](#)
- [CameraDefs Class](#)
- [Camera List Class](#)
- [CameraPtr Class](#)
- [ChunkData Class](#)
- [Exception Class](#)
- [Image Class](#)
- [ImagePtr Class](#)
- [ImageStatistics Class](#)
- [Interface Class](#)
- [InterfaceList Class](#)
- [InterfacePtr Class](#)
- [Spinnaker Video Class](#)
- [System Class](#)
- [SystemPtr Class](#)
- [Camera Base Interface Class](#)
- [IChunkData Class](#)
- [IImage Class](#)
- [IImageStatistics Class](#)
- [IInterface Class](#)
- [IInterfaceList Class](#)
- [ISystem Class](#)

## Classes

- class [BasePtr< T, B >](#)  
*The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.*
- class [Camera](#)  
*The camera object class.*
- class [CameraBase](#)  
*The base class for the camera object.*
- class [CameraList](#)  
*Used to hold a list of camera objects.*
- class [CameraPtr](#)  
*A reference tracked pointer to a camera object.*
- class [ChunkData](#)  
*The chunk data which contains additional information about an image.*
- class [Exception](#)  
*The [Exception](#) object represents an error that is returned from the library.*
- class [Image](#)  
*The image object class.*
- class [ImagePtr](#)  
*A reference tracked pointer to an image object.*
- class [ImageStatistics](#)  
*Represents image statistics for an image.*
- class [Interface](#)  
*An interface object which holds a list of cameras.*
- class [InterfaceList](#)  
*A list of the available interfaces on the system.*
- class [InterfacePtr](#)

- A reference tracked pointer to the interface object.*
- class [SpinVideo](#)  
*Provides the functionality for the user to record images to an AVI/MP4 file.*
- class [System](#)  
*The system object is used to retrieve the list of interfaces and cameras available.*
- class [SystemPtr](#)  
*A reference tracked pointer to a system object.*
- class [ICameraBase](#)  
*The interface file for base class for the camera object.*
- class [ICameraList](#)  
*Used to hold a list of camera objects.*
- class [IChunkData](#)  
*The [Interface](#) file for [ChunkData](#).*
- class [IImage](#)  
*The interface file for [Image](#).*
- class [IImageStatistics](#)  
*The interface file for image statistics.*
- class [IInterface](#)  
*The interface file for [Interface](#).*
- class [IInterfaceList](#)  
*The interface file for [InterfaceList](#) class.*
- class [ISystem](#)  
*The interface file for [System](#).*

### 8.3.1 Detailed Description

## 8.4 AVI Recorder Class

Collaboration diagram for AVI Recorder Class:



### Functions

- class [DEPRECATED\\_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER\_API A↔ VIRECORDER

*Provides the functionality for the user to record images to an AVI file.*

#### 8.4.1 Detailed Description

#### 8.4.2 Function Documentation

##### 8.4.2.1 class Spinnaker::DEPRECATED\_CLASS ( "AVIRecorder is *deprecated*, use SpinVideo instead." )

Provides the functionality for the user to record images to an AVI file.

NOTE: This class is deprecated and replaced by SpinVideo. Refer to [SpinVideo.h](#) instead. Default constructor.

Default destructor.

Open an AVI file in preparation for writing Images to disk. The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

##### Parameters

<i>pFileName</i>	The filename of the AVI file.
<i>pOption</i>	Options to apply to the AVI file.

##### See also

AVIClose()

Open an MJPEG AVI file in preparation for writing Images to disk. The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

##### Parameters

<i>pFileName</i>	The filename of the AVI file.
<i>pOption</i>	MJPEG options to apply to the AVI file.

**See also**

AVIClose()  
MJPGOption

Open an H264 MP4 file in preparation for writing Images to disk. The size of MP4 files is limited to 2GB. The filenames are automatically generated using the filename specified.

**Parameters**

<i>pFileName</i>	The filename of the MP4 file.
<i>pOption</i>	H264 options to apply to the MP4 file.

**See also**

AVIClose()  
H264Option

Append an image to the AVI/MP4 file.

**Parameters**

<i>pImage</i>	The image to append.
---------------	----------------------

Close the AVI/MP4 file.

**See also**

AVIOpen()

Set the maximum file size (in megabytes) of a AVI/MP4 file. A new AVI/MP4 file is created automatically when file size limit is reached. Setting a maximum size of 0 indicates no limit on file size.

**Parameters**

<i>size</i>	The maximum AVI file size in MB.
-------------	----------------------------------

**See also**

AVIAppend( ImagePtr pImage)

## 8.5 BasePtr Class

Collaboration diagram for BasePtr Class:



### Classes

- class `BasePtr< T, B >`

*The base class of the `SystemPtr`, `CameraPtr`, `InterfacePtr`, `ImagePtr` and `LoggingEventDataPtr` objects.*

#### 8.5.1 Detailed Description

## 8.6 Camera Class

Collaboration diagram for Camera Class:



### Classes

- class [Camera](#)  
*The camera object class.*

#### 8.6.1 Detailed Description



## 8.7 Camera Base Class

Collaboration diagram for Camera Base Class:



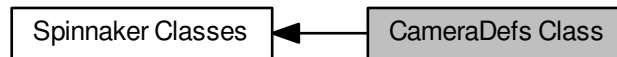
### Classes

- class [CameraBase](#)  
*The base class for the camera object.*

#### 8.7.1 Detailed Description

## 8.8 CameraDefs Class

Collaboration diagram for CameraDefs Class:



### Enumerations

- enum `LUTSelectorEnums` {  
`LUTSelector_LUT1`,  
`NUM_LUTSELECTOR` }

*The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.*

- enum `ExposureModeEnums` {  
`ExposureMode_Timed`,  
`ExposureMode_TriggerWidth`,  
`NUM_EXPOSUREMODE` }
- enum `AcquisitionModeEnums` {  
`AcquisitionMode_Continuous`,  
`AcquisitionMode_SingleFrame`,  
`AcquisitionMode_MultiFrame`,  
`NUM_ACQUISITIONMODE` }
- enum `TriggerSourceEnums` {  
`TriggerSource_Software`,  
`TriggerSource_Line0`,  
`TriggerSource_Line1`,  
`TriggerSource_Line2`,  
`TriggerSource_Line3`,  
`TriggerSource_UserOutput0`,  
`TriggerSource_UserOutput1`,  
`TriggerSource_UserOutput2`,  
`TriggerSource_UserOutput3`,  
`TriggerSource_Counter0Start`,  
`TriggerSource_Counter1Start`,  
`TriggerSource_Counter0End`,  
`TriggerSource_Counter1End`,  
`TriggerSource_LogicBlock0`,  
`TriggerSource_LogicBlock1`,  
`TriggerSource_Action0`,  
`NUM_TRIGGERSOURCE` }
- enum `TriggerActivationEnums` {  
`TriggerActivation_LevelLow`,  
`TriggerActivation_LevelHigh`,  
`TriggerActivation_FallingEdge`,  
`TriggerActivation_RisingEdge`,  
`TriggerActivation_AnyEdge`,  
`NUM_TRIGGERACTIVATION` }

- enum `SensorShutterModeEnums` {  
    `SensorShutterMode_Global`,  
    `SensorShutterMode_Rolling`,  
    `SensorShutterMode_GlobalReset`,  
    `NUM_SENSORSHUTTERMODE` }
- enum `TriggerModeEnums` {  
    `TriggerMode_Off`,  
    `TriggerMode_On`,  
    `NUM_TRIGGERMODE` }
- enum `TriggerOverlapEnums` {  
    `TriggerOverlap_Off`,  
    `TriggerOverlap_ReadOut`,  
    `TriggerOverlap_PreviousFrame`,  
    `NUM_TRIGGEROVERLAP` }
- enum `TriggerSelectorEnums` {  
    `TriggerSelector_AcquisitionStart`,  
    `TriggerSelector_FrameStart`,  
    `TriggerSelector_FrameBurstStart`,  
    `NUM_TRIGGERSELECTOR` }
- enum `ExposureAutoEnums` {  
    `ExposureAuto_Off`,  
    `ExposureAuto_Once`,  
    `ExposureAuto_Continuous`,  
    `NUM_EXPOSUREAUTO` }
- enum `EventSelectorEnums` {  
    `EventSelector_Error`,  
    `EventSelector_ExposureEnd`,  
    `EventSelector_SerialPortReceive`,  
    `NUM_EVENTSELECTOR` }
- enum `EventNotificationEnums` {  
    `EventNotification_On`,  
    `EventNotification_Off`,  
    `NUM_EVENTNOTIFICATION` }
- enum `LogicBlockSelectorEnums` {  
    `LogicBlockSelector_LogicBlock0`,  
    `LogicBlockSelector_LogicBlock1`,  
    `NUM_LOGICBLOCKSELECTOR` }
- enum `LogicBlockLUTInputActivationEnums` {  
    `LogicBlockLUTInputActivation_LevelLow`,  
    `LogicBlockLUTInputActivation_LevelHigh`,  
    `LogicBlockLUTInputActivation_FallingEdge`,  
    `LogicBlockLUTInputActivation_RisingEdge`,  
    `LogicBlockLUTInputActivation_AnyEdge`,  
    `NUM_LOGICBLOCKLUTINPUTACTIVATION` }
- enum `LogicBlockLUTInputSelectorEnums` {  
    `LogicBlockLUTInputSelector_Input0`,  
    `LogicBlockLUTInputSelector_Input1`,  
    `LogicBlockLUTInputSelector_Input2`,  
    `LogicBlockLUTInputSelector_Input3`,  
    `NUM_LOGICBLOCKLUTINPUTSELECTOR` }
- enum `LogicBlockLUTInputSourceEnums` {

```

LogicBlockLUTInputSource_Zero,
LogicBlockLUTInputSource_Line0,
LogicBlockLUTInputSource_Line1,
LogicBlockLUTInputSource_Line2,
LogicBlockLUTInputSource_Line3,
LogicBlockLUTInputSource_UserOutput0,
LogicBlockLUTInputSource_UserOutput1,
LogicBlockLUTInputSource_UserOutput2,
LogicBlockLUTInputSource_UserOutput3,
LogicBlockLUTInputSource_Counter0Start,
LogicBlockLUTInputSource_Counter1Start,
LogicBlockLUTInputSource_Counter0End,
LogicBlockLUTInputSource_Counter1End,
LogicBlockLUTInputSource_LogicBlock0,
LogicBlockLUTInputSource_LogicBlock1,
LogicBlockLUTInputSource_ExposureStart,
LogicBlockLUTInputSource_ExposureEnd,
LogicBlockLUTInputSource_FrameTriggerWait,
LogicBlockLUTInputSource_AcquisitionActive,
NUM_LOGICBLOCKLUTINPUTSOURCE }

• enum LogicBlockLUTSelectorEnums {
    LogicBlockLUTSelector_Value,
    LogicBlockLUTSelector_Enable,
    NUM_LOGICBLOCKLUTSELECTOR }

• enum ColorTransformationSelectorEnums {
    ColorTransformationSelector_RGBtoRGB,
    ColorTransformationSelector_RGBtoYUV,
    NUM_COLORTRANSFORMATIONSELECTOR }

• enum RgbTransformLightSourceEnums {
    RgbTransformLightSource_General,
    RgbTransformLightSource_Tungsten2800K,
    RgbTransformLightSource_WarmFluorescent3000K,
    RgbTransformLightSource_CoolFluorescent4000K,
    RgbTransformLightSource_Daylight5000K,
    RgbTransformLightSource_Cloudy6500K,
    RgbTransformLightSource_Shade8000K,
    RgbTransformLightSource_Custom,
    NUM_RGBTRANSFORMLIGHTSOURCE }

• enum ColorTransformationValueSelectorEnums {
    ColorTransformationValueSelector_Gain00,
    ColorTransformationValueSelector_Gain01,
    ColorTransformationValueSelector_Gain02,
    ColorTransformationValueSelector_Gain10,
    ColorTransformationValueSelector_Gain11,
    ColorTransformationValueSelector_Gain12,
    ColorTransformationValueSelector_Gain20,
    ColorTransformationValueSelector_Gain21,
    ColorTransformationValueSelector_Gain22,
    ColorTransformationValueSelector_Offset0,
    ColorTransformationValueSelector_Offset1,
    ColorTransformationValueSelector_Offset2,
    NUM_COLORTRANSFORMATIONVALUESELECTOR }

• enum DeviceRegistersEndiannessEnums {
    DeviceRegistersEndianness_Little,
    DeviceRegistersEndianness_Big,
    NUM_DEVICEREGISTERSENDIANNES }

• enum DeviceScanTypeEnums {
    DeviceScanType_Areascan,

```

- ```
NUM_DEVICESCANTYPE }
```
- enum DeviceCharacterSetEnums {  
DeviceCharacterSet\_UTF8,  
DeviceCharacterSet\_ASCII,  
NUM\_DEVICECHARACTERSET }
  - enum DeviceTLTypeEnums {  
DeviceTLType\_GigEVision,  
DeviceTLType\_CameraLink,  
DeviceTLType\_CameraLinkHS,  
DeviceTLType\_CoaXPress,  
DeviceTLType\_USB3Vision,  
DeviceTLType\_Custom,  
NUM\_DEVICETLTYPE }
  - enum DevicePowerSupplySelectorEnums {  
DevicePowerSupplySelector\_External,  
NUM\_DEVICEPOWERSUPPLYSELECTOR }
  - enum DeviceTemperatureSelectorEnums {  
DeviceTemperatureSelector\_Sensor,  
NUM\_DEVICETEMPERATURESELECTOR }
  - enum DeviceIndicatorModeEnums {  
DeviceIndicatorMode\_Inactive,  
DeviceIndicatorMode\_Active,  
DeviceIndicatorMode\_ErrorStatus,  
NUM\_DEVICEINDICATORMODE }
  - enum AutoExposureControlPriorityEnums {  
AutoExposureControlPriority\_Gain,  
AutoExposureControlPriority\_ExposureTime,  
NUM\_AUTOEXPOSURECONTROLPRIORITY }
  - enum AutoExposureMeteringModeEnums {  
AutoExposureMeteringMode\_Average,  
AutoExposureMeteringMode\_Spot,  
AutoExposureMeteringMode\_Partial,  
AutoExposureMeteringMode\_CenterWeighted,  
AutoExposureMeteringMode\_HistogramPeak,  
NUM\_AUTOEXPOSUREMETERINGMODE }
  - enum BalanceWhiteAutoProfileEnums {  
BalanceWhiteAutoProfile\_Indoor,  
BalanceWhiteAutoProfile\_Outdoor,  
NUM\_BALANCEWHITEAUTOPROFILE }
  - enum AutoAlgorithmSelectorEnums {  
AutoAlgorithmSelector\_Awb,  
AutoAlgorithmSelector\_Ae,  
NUM\_AUTOALGORITHMSELECTOR }
  - enum AutoExposureTargetGreyValueAutoEnums {  
AutoExposureTargetGreyValueAuto\_Off,  
AutoExposureTargetGreyValueAuto\_Continuous,  
NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO }
  - enum AutoExposureLightingModeEnums {  
AutoExposureLightingMode\_AutoDetect,  
AutoExposureLightingMode\_Backlight,  
AutoExposureLightingMode\_Frontlight,  
AutoExposureLightingMode\_Normal,  
NUM\_AUTOEXPOSURELIGHTINGMODE }
  - enum GevIEEE1588StatusEnums {

```

    GevIEEE1588Status_Initializing,
    GevIEEE1588Status_Faulty,
    GevIEEE1588Status_Disabled,
    GevIEEE1588Status_Listening,
    GevIEEE1588Status_PreMaster,
    GevIEEE1588Status_Master,
    GevIEEE1588Status_Passive,
    GevIEEE1588Status_Uncalibrated,
    GevIEEE1588Status_Slave,
    NUM_GEVIEEE1588STATUS }

• enum GevIEEE1588ModeEnums {
    GevIEEE1588Mode_Auto,
    GevIEEE1588Mode_SlaveOnly,
    NUM_GEVIEEE1588MODE }

• enum GevIEEE1588ClockAccuracyEnums {
    GevIEEE1588ClockAccuracy_Unknown,
    NUM_GEVIEEE1588CLOCKACCURACY }

• enum GevCCPEnums {
    GevCCP_OpenAccess,
    GevCCP_ExclusiveAccess,
    GevCCP_ControlAccess,
    NUM_GEVCCP }

• enum GevSupportedOptionSelectorEnums {
    GevSupportedOptionSelector_UserDefinedName,
    GevSupportedOptionSelector_SerialNumber,
    GevSupportedOptionSelector_HeartbeatDisable,
    GevSupportedOptionSelector_LinkSpeed,
    GevSupportedOptionSelector_CCPApplicationSocket,
    GevSupportedOptionSelector_ManifestTable,
    GevSupportedOptionSelector_TestData,
    GevSupportedOptionSelector_DiscoveryAckDelay,
    GevSupportedOptionSelector_DiscoveryAckDelayWritable,
    GevSupportedOptionSelector_ExtendedStatusCodes,
    GevSupportedOptionSelector_Action,
    GevSupportedOptionSelector_PendingAck,
    GevSupportedOptionSelector_EventData,
    GevSupportedOptionSelector_Event,
    GevSupportedOptionSelector_PacketResend,
    GevSupportedOptionSelector_WriteMem,
    GevSupportedOptionSelector_CommandsConcatenation,
    GevSupportedOptionSelector_IPConfigurationLLA,
    GevSupportedOptionSelector_IPConfigurationDHCP,
    GevSupportedOptionSelector_IPConfigurationPersistentIP,
    GevSupportedOptionSelector_StreamChannelSourceSocket,
    GevSupportedOptionSelector_MessageChannelSourceSocket,
    NUM_GEVSUPPORTEDOPTIONSELECTOR }

• enum BlackLevelSelectorEnums {
    BlackLevelSelector_All,
    BlackLevelSelector_Analog,
    BlackLevelSelector_Digital,
    NUM_BLACKLEVELSELECTOR }

• enum BalanceWhiteAutoEnums {
    BalanceWhiteAuto_Off,
    BalanceWhiteAuto_Once,
    BalanceWhiteAuto_Continuous,
    NUM_BALANCEWHITEAUTO }

• enum GainAutoEnums {

```

```
GainAuto_Off,  
GainAuto_Once,  
GainAuto_Continuous,  
NUM_GAINAUTO }  
  
• enum BalanceRatioSelectorEnums {  
    BalanceRatioSelector_Red,  
    BalanceRatioSelector_Blue,  
    NUM_BALANCERATIOSELECTOR }  
  
• enum GainSelectorEnums {  
    GainSelector_All,  
    NUM_GAINSELECTOR }  
  
• enum DefectCorrectionModeEnums {  
    DefectCorrectionMode_Average,  
    DefectCorrectionMode_Highlight,  
    DefectCorrectionMode_Zero,  
    NUM_DEFECTCORRECTIONMODE }  
  
• enum UserSetSelectorEnums {  
    UserSetSelector_Default,  
    UserSetSelector_UserSet0,  
    UserSetSelector_UserSet1,  
    NUM_USERSETSELECTOR }  
  
• enum UserSetDefaultEnums {  
    UserSetDefault_Default,  
    UserSetDefault_UserSet0,  
    UserSetDefault_UserSet1,  
    NUM_USERSETDEFAULT }  
  
• enum SerialPortBaudRateEnums {  
    SerialPortBaudRate_Baud300,  
    SerialPortBaudRate_Baud600,  
    SerialPortBaudRate_Baud1200,  
    SerialPortBaudRate_Baud2400,  
    SerialPortBaudRate_Baud4800,  
    SerialPortBaudRate_Baud9600,  
    SerialPortBaudRate_Baud14400,  
    SerialPortBaudRate_Baud19200,  
    SerialPortBaudRate_Baud38400,  
    SerialPortBaudRate_Baud57600,  
    SerialPortBaudRate_Baud115200,  
    SerialPortBaudRate_Baud230400,  
    SerialPortBaudRate_Baud460800,  
    SerialPortBaudRate_Baud921600,  
    NUM_SERIALPORTBAUDRATE }  
  
• enum SerialPortParityEnums {  
    SerialPortParity_None,  
    SerialPortParity_Odd,  
    SerialPortParity_Even,  
    SerialPortParity_Mark,  
    SerialPortParity_Space,  
    NUM_SERIALPORTPARITY }  
  
• enum SerialPortSelectorEnums {  
    SerialPortSelector_SerialPort0,  
    NUM_SERIALPORTSELECTOR }  
  
• enum SerialPortStopBitsEnums {  
    SerialPortStopBits_Bits1,  
    SerialPortStopBits_Bits1AndAHalf,  
    SerialPortStopBits_Bits2,  
    NUM_SERIALPORTSTOPBITS }
```

- enum `SerialPortSourceEnums` {  
    `SerialPortSource_Line0`,  
    `SerialPortSource_Line1`,  
    `SerialPortSource_Line2`,  
    `SerialPortSource_Line3`,  
    `SerialPortSource_Off`,  
    `NUM_SERIALPORTSOURCE` }
- enum `SequencerModeEnums` {  
    `SequencerMode_Off`,  
    `SequencerMode_On`,  
    `NUM_SEQUENCERMODE` }
- enum `SequencerConfigurationValidEnums` {  
    `SequencerConfigurationValid_No`,  
    `SequencerConfigurationValid_Yes`,  
    `NUM_SEQUENCERCONFIGURATIONVALID` }
- enum `SequencerSetValidEnums` {  
    `SequencerSetValid_No`,  
    `SequencerSetValid_Yes`,  
    `NUM_SEQUENCERSETVALID` }
- enum `SequencerTriggerActivationEnums` {  
    `SequencerTriggerActivation_RisingEdge`,  
    `SequencerTriggerActivation_FallingEdge`,  
    `SequencerTriggerActivation_AnyEdge`,  
    `SequencerTriggerActivation_LevelHigh`,  
    `SequencerTriggerActivation_LevelLow`,  
    `NUM_SEQUENCERTRIGGERACTIVATION` }
- enum `SequencerConfigurationModeEnums` {  
    `SequencerConfigurationMode_Off`,  
    `SequencerConfigurationMode_On`,  
    `NUM_SEQUENCERCONFIGURATIONMODE` }
- enum `SequencerTriggerSourceEnums` {  
    `SequencerTriggerSource_Off`,  
    `SequencerTriggerSource_FrameStart`,  
    `NUM_SEQUENCERTRIGGERSOURCE` }
- enum `TransferQueueModeEnums` {  
    `TransferQueueMode_FirstInFirstOut`,  
    `NUM_TRANSFERQUEUEMODE` }
- enum `TransferOperationModeEnums` {  
    `TransferOperationMode_Continuous`,  
    `TransferOperationMode_MultiBlock`,  
    `NUM_TRANSFEROPERATIONMODE` }
- enum `TransferControlModeEnums` {  
    `TransferControlMode_Basic`,  
    `TransferControlMode_Automatic`,  
    `TransferControlMode_UserControlled`,  
    `NUM_TRANSFERCONTROLMODE` }
- enum `ChunkGainSelectorEnums` {  
    `ChunkGainSelector_All`,  
    `ChunkGainSelector_Red`,  
    `ChunkGainSelector_Green`,  
    `ChunkGainSelector_Blue`,  
    `NUM_CHUNKGAINSELECTOR` }
- enum `ChunkSelectorEnums` {



```

    ChunkSelector_Image,
    ChunkSelector_CRC,
    ChunkSelector_FrameID,
    ChunkSelector_OffsetX,
    ChunkSelector_OffsetY,
    ChunkSelector_Width,
    ChunkSelector_Height,
    ChunkSelector_ExposureTime,
    ChunkSelector_Gain,
    ChunkSelector_BlackLevel,
    ChunkSelector_PixelFormat,
    ChunkSelector_Timestamp,
    ChunkSelector_SequencerSetActive,
    ChunkSelector_SerialData,
    ChunkSelector_ExposureEndLineStatusAll,
    NUM_CHUNKSELECTOR }

• enum ChunkBlackLevelSelectorEnums {
    ChunkBlackLevelSelector_All,
    NUM_CHUNKBLACKLEVELSELECTOR }

• enum ChunkPixelFormatEnums {
    ChunkPixelFormat_Mono8,
    ChunkPixelFormat_Mono12Packed,
    ChunkPixelFormat_Mono16,
    ChunkPixelFormat_RGB8Packed,
    ChunkPixelFormat_YUV422Packed,
    ChunkPixelFormat_BayerGR8,
    ChunkPixelFormat_BayerRG8,
    ChunkPixelFormat_BayerGB8,
    ChunkPixelFormat_BayerBG8,
    ChunkPixelFormat_YCbCr601_422_8_CbYCrY,
    NUM_CHUNKPIXELFORMAT }

• enum FileOperationStatusEnums {
    FileOperationStatus_Success,
    FileOperationStatus_Failure,
    FileOperationStatus_Overflow,
    NUM_FILEOPERATIONSTATUS }

• enum FileOpenModeEnums {
    FileOpenMode_Read,
    FileOpenMode_Write,
    FileOpenMode_ReadWrite,
    NUM_FILEOPENMODE }

• enum FileOperationSelectorEnums {
    FileOperationSelector_Open,
    FileOperationSelector_Close,
    FileOperationSelector_Read,
    FileOperationSelector_Write,
    FileOperationSelector_Delete,
    NUM_FILEOPERATIONSELECTOR }

• enum FileSelectorEnums {
    FileSelector_UserSetDefault,
    FileSelector_UserSet0,
    FileSelector_UserSet1,
    FileSelector_UserFile1,
    FileSelector_SerialPort0,
    NUM_FILESELECTOR }

• enum BinningSelectorEnums {

```

```
BinningSelector_All,  
BinningSelector_Sensor,  
BinningSelector_ISP,  
NUM_BINNINGSELECTOR }  
• enum TestPatternGeneratorSelectorEnums {  
    TestPatternGeneratorSelector_Sensor,  
    TestPatternGeneratorSelector_PipelineStart,  
    NUM_TESTPATTERNGENERATORSELECTOR }  
• enum TestPatternEnums {  
    TestPattern_Off,  
    TestPattern_Increment,  
    TestPattern_SensorTestPattern,  
    NUM_TESTPATTERN }  
• enum PixelColorFilterEnums {  
    PixelColorFilter_None,  
    PixelColorFilter_BayerRG,  
    PixelColorFilter_BayerGB,  
    PixelColorFilter_BayerGR,  
    PixelColorFilter_BayerBG,  
    NUM_PIXELCOLORFILTER }  
• enum AdcBitDepthEnums {  
    AdcBitDepth_Bit8,  
    AdcBitDepth_Bit10,  
    AdcBitDepth_Bit12,  
    AdcBitDepth_Bit14,  
    NUM_ADCBITDEPTH }  
• enum DecimationHorizontalModeEnums {  
    DecimationHorizontalMode_Discard,  
    NUM_DECIMATIONHORIZONTALMODE }  
• enum BinningVerticalModeEnums {  
    BinningVerticalMode_Sum,  
    BinningVerticalMode_Average,  
    NUM_BINNINGVERTICALMODE }  
• enum PixelSizeEnums {  
    PixelSize_Bpp1,  
    PixelSize_Bpp2,  
    PixelSize_Bpp4,  
    PixelSize_Bpp8,  
    PixelSize_Bpp10,  
    PixelSize_Bpp12,  
    PixelSize_Bpp14,  
    PixelSize_Bpp16,  
    PixelSize_Bpp20,  
    PixelSize_Bpp24,  
    PixelSize_Bpp30,  
    PixelSize_Bpp32,  
    PixelSize_Bpp36,  
    PixelSize_Bpp48,  
    PixelSize_Bpp64,  
    PixelSize_Bpp96,  
    NUM_PIXELSIZE }  
• enum DecimationSelectorEnums {  
    DecimationSelector_All,  
    DecimationSelector_Sensor,  
    NUM_DECIMATIONSELECTOR }  
• enum ImageCompressionModeEnums {  
    ImageCompressionMode_Off,  
    ImageCompressionMode_Lossless,
```

```
NUM_IMAGECOMPRESSIONMODE }
```

- enum BinningHorizontalModeEnums {  
    BinningHorizontalMode\_Sum,  
    BinningHorizontalMode\_Average,  
    NUM\_BINNINGHORIZONTALMODE }

- enum PixelFormatEnums {

PixelFormat\_Mono8,  
PixelFormat\_Mono16,  
PixelFormat\_RGB8Packed,  
PixelFormat\_BayerGR8,  
PixelFormat\_BayerRG8,  
PixelFormat\_BayerGB8,  
PixelFormat\_BayerBG8,  
PixelFormat\_BayerGR16,  
PixelFormat\_BayerRG16,  
PixelFormat\_BayerGB16,  
PixelFormat\_BayerBG16,  
PixelFormat\_Mono12Packed,  
PixelFormat\_BayerGR12Packed,  
PixelFormat\_BayerRG12Packed,  
PixelFormat\_BayerGB12Packed,  
PixelFormat\_BayerBG12Packed,  
PixelFormat\_YUV411Packed,  
PixelFormat\_YUV422Packed,  
PixelFormat\_YUV444Packed,  
PixelFormat\_Mono12p,  
PixelFormat\_BayerGR12p,  
PixelFormat\_BayerRG12p,  
PixelFormat\_BayerGB12p,  
PixelFormat\_BayerBG12p,  
PixelFormat\_YCbCr8,  
PixelFormat\_YCbCr422\_8,  
PixelFormat\_YCbCr411\_8,  
PixelFormat\_BGR8,  
PixelFormat\_BGRa8,  
PixelFormat\_Mono10Packed,  
PixelFormat\_BayerGR10Packed,  
PixelFormat\_BayerRG10Packed,  
PixelFormat\_BayerGB10Packed,  
PixelFormat\_BayerBG10Packed,  
PixelFormat\_Mono10p,  
PixelFormat\_BayerGR10p,  
PixelFormat\_BayerRG10p,  
PixelFormat\_BayerGB10p,  
PixelFormat\_BayerBG10p,  
PixelFormat\_Mono1p,  
PixelFormat\_Mono2p,  
PixelFormat\_Mono4p,  
PixelFormat\_Mono8s,  
PixelFormat\_Mono10,  
PixelFormat\_Mono12,  
PixelFormat\_Mono14,  
PixelFormat\_BayerBG10,  
PixelFormat\_BayerBG12,  
PixelFormat\_BayerGB10,  
PixelFormat\_BayerGB12,  
PixelFormat\_BayerGR10,  
PixelFormat\_BayerGR12,  
PixelFormat\_BayerRG10,  
PixelFormat\_BayerRG12,  
PixelFormat\_RGBa8,  
PixelFormat\_RGBa10,  
PixelFormat\_RGBa10p,  
PixelFormat\_RGBa12,  
PixelFormat\_RGBa12p,  
PixelFormat\_RGBa14,  
PixelFormat\_RGBa16,  
PixelFormat\_RGB8,  
PixelFormat\_RGB8\_Planar,  
PixelFormat\_RGB10,  
PixelFormat\_RGB10\_Planar,

- ```
NUM_PIXELFORMAT }
```
- enum DecimationVerticalModeEnums {  
DecimationVerticalMode\_Discard,  
NUM\_DECIMATIONVERTICALMODE }
  - enum LineModeEnums {  
LineMode\_Input,  
LineMode\_Output,  
NUM\_LINEMODE }
  - enum LineSourceEnums {  
LineSource\_Off,  
LineSource\_Line0,  
LineSource\_Line1,  
LineSource\_Line2,  
LineSource\_Line3,  
LineSource\_UserOutput0,  
LineSource\_UserOutput1,  
LineSource\_UserOutput2,  
LineSource\_UserOutput3,  
LineSource\_Counter0Active,  
LineSource\_Counter1Active,  
LineSource\_LogicBlock0,  
LineSource\_LogicBlock1,  
LineSource\_ExposureActive,  
LineSource\_FrameTriggerWait,  
LineSource\_SerialPort0,  
LineSource\_PPSSignal,  
LineSource\_AllPixel,  
LineSource\_AnyPixel,  
NUM\_LINESOURCE }
  - enum LineInputFilterSelectorEnums {  
LineInputFilterSelector\_Deglintch,  
LineInputFilterSelector\_Debounce,  
NUM\_LINEINPUTFILTERSELECTOR }
  - enum UserOutputSelectorEnums {  
UserOutputSelector\_UserOutput0,  
UserOutputSelector\_UserOutput1,  
UserOutputSelector\_UserOutput2,  
UserOutputSelector\_UserOutput3,  
NUM\_USEROUTPUTSELECTOR }
  - enum LineFormatEnums {  
LineFormat\_NoConnect,  
LineFormat\_TriState,  
LineFormat\_TTL,  
LineFormat\_LVDS,  
LineFormat\_RS422,  
LineFormat\_OptoCoupled,  
LineFormat\_OpenDrain,  
NUM\_LINEFORMAT }
  - enum LineSelectorEnums {  
LineSelector\_Line0,  
LineSelector\_Line1,  
LineSelector\_Line2,  
LineSelector\_Line3,  
NUM\_LINESELECTOR }
  - enum ExposureActiveModeEnums {  
ExposureActiveMode\_Line1,  
ExposureActiveMode\_AnyPixels,  
ExposureActiveMode\_AllPixels,

```

NUM_EXPOSUREACTIVEMODE }
• enum CounterTriggerActivationEnums {
    CounterTriggerActivation_LevelLow,
    CounterTriggerActivation_LevelHigh,
    CounterTriggerActivation_FallingEdge,
    CounterTriggerActivation_RisingEdge,
    CounterTriggerActivation_AnyEdge,
    NUM_COUNTERTRIGGERACTIVATION }
• enum CounterSelectorEnums {
    CounterSelector_Counter0,
    CounterSelector_Counter1,
    NUM_COUNTERSELECTOR }
• enum CounterStatusEnums {
    CounterStatus_CounterIdle,
    CounterStatus_CounterTriggerWait,
    CounterStatus_CounterActive,
    CounterStatus_CounterCompleted,
    CounterStatus_CounterOverflow,
    NUM_COUNTERSTATUS }
• enum CounterTriggerSourceEnums {
    CounterTriggerSource_Off,
    CounterTriggerSource_Line0,
    CounterTriggerSource_Line1,
    CounterTriggerSource_Line2,
    CounterTriggerSource_Line3,
    CounterTriggerSource_UserOutput0,
    CounterTriggerSource_UserOutput1,
    CounterTriggerSource_UserOutput2,
    CounterTriggerSource_UserOutput3,
    CounterTriggerSource_Counter0Start,
    CounterTriggerSource_Counter1Start,
    CounterTriggerSource_Counter0End,
    CounterTriggerSource_Counter1End,
    CounterTriggerSource_LogicBlock0,
    CounterTriggerSource_LogicBlock1,
    CounterTriggerSource_ExposureStart,
    CounterTriggerSource_ExposureEnd,
    CounterTriggerSource_FrameTriggerWait,
    NUM_COUNTERTRIGGERSOURCE }
• enum CounterResetSourceEnums {
    CounterResetSource_Off,
    CounterResetSource_Line0,
    CounterResetSource_Line1,
    CounterResetSource_Line2,
    CounterResetSource_Line3,
    CounterResetSource_UserOutput0,
    CounterResetSource_UserOutput1,
    CounterResetSource_UserOutput2,
    CounterResetSource_UserOutput3,
    CounterResetSource_Counter0Start,
    CounterResetSource_Counter1Start,
    CounterResetSource_Counter0End,
    CounterResetSource_Counter1End,
    CounterResetSource_LogicBlock0,
    CounterResetSource_LogicBlock1,
    CounterResetSource_ExposureStart,
    CounterResetSource_ExposureEnd,
    CounterResetSource_FrameTriggerWait,

```

```

    NUM_COUNTERRESETSOURCE }
    • enum CounterEventSourceEnums {
        CounterEventSource_Off,
        CounterEventSource_MHzTick,
        CounterEventSource_Line0,
        CounterEventSource_Line1,
        CounterEventSource_Line2,
        CounterEventSource_Line3,
        CounterEventSource_UserOutput0,
        CounterEventSource_UserOutput1,
        CounterEventSource_UserOutput2,
        CounterEventSource_UserOutput3,
        CounterEventSource_Counter0Start,
        CounterEventSource_Counter1Start,
        CounterEventSource_Counter0End,
        CounterEventSource_Counter1End,
        CounterEventSource_LogicBlock0,
        CounterEventSource_LogicBlock1,
        CounterEventSource_ExposureStart,
        CounterEventSource_ExposureEnd,
        CounterEventSource_FrameTriggerWait,
        NUM_COUNTEREVENTSOURCE }
    • enum CounterEventActivationEnums {
        CounterEventActivation_LevelLow,
        CounterEventActivation_LevelHigh,
        CounterEventActivation_FallingEdge,
        CounterEventActivation_RisingEdge,
        CounterEventActivation_AnyEdge,
        NUM_COUNTEREVENTACTIVATION }
    • enum CounterResetActivationEnums {
        CounterResetActivation_LevelLow,
        CounterResetActivation_LevelHigh,
        CounterResetActivation_FallingEdge,
        CounterResetActivation_RisingEdge,
        CounterResetActivation_AnyEdge,
        NUM_COUNTERRESETACTIVATION }
    • enum DeviceTypeEnums {
        DeviceType_Transmitter,
        DeviceType_Receiver,
        DeviceType_Transceiver,
        DeviceType_Peripheral,
        NUM_DEVICETYPE }
    • enum DeviceConnectionStatusEnums {
        DeviceConnectionStatus_Active,
        DeviceConnectionStatus_Inactive,
        NUM_DEVICECONNECTIONSTATUS }
    • enum DeviceLinkThroughputLimitModeEnums {
        DeviceLinkThroughputLimitMode_On,
        DeviceLinkThroughputLimitMode_Off,
        NUM_DEVICELINKTHROUGHPUTLIMITMODE }
    • enum DeviceLinkHeartbeatModeEnums {
        DeviceLinkHeartbeatMode_On,
        DeviceLinkHeartbeatMode_Off,
        NUM_DEVICELINKHEARTBEATMODE }
    • enum DeviceStreamChannelTypeEnums {
        DeviceStreamChannelType_Transmitter,
        DeviceStreamChannelType_Receiver,
        NUM_DEVICESTREAMCHANNELTYPE }

```

- enum [DeviceStreamChannelEndiannessEnums](#) {  
    [DeviceStreamChannelEndianness\\_Big](#),  
    [DeviceStreamChannelEndianness\\_Little](#),  
    [NUM\\_DEVICESTREAMCHANNELENDIANNESS](#) }
- enum [DeviceClockSelectorEnums](#) {  
    [DeviceClockSelector\\_Sensor](#),  
    [DeviceClockSelector\\_SensorDigitization](#),  
    [DeviceClockSelector\\_CameraLink](#),  
    [NUM\\_DEVICECLOCKSELECTOR](#) }
- enum [DeviceSerialPortSelectorEnums](#) {  
    [DeviceSerialPortSelector\\_CameraLink](#),  
    [NUM\\_DEVICESERIALPORTSELECTOR](#) }
- enum [DeviceSerialPortBaudRateEnums](#) {  
    [DeviceSerialPortBaudRate\\_Baud9600](#),  
    [DeviceSerialPortBaudRate\\_Baud19200](#),  
    [DeviceSerialPortBaudRate\\_Baud38400](#),  
    [DeviceSerialPortBaudRate\\_Baud57600](#),  
    [DeviceSerialPortBaudRate\\_Baud115200](#),  
    [DeviceSerialPortBaudRate\\_Baud230400](#),  
    [DeviceSerialPortBaudRate\\_Baud460800](#),  
    [DeviceSerialPortBaudRate\\_Baud921600](#),  
    [NUM\\_DEVICESERIALPORTBAUDRATE](#) }
- enum [SensorTapsEnums](#) {  
    [SensorTaps\\_One](#),  
    [SensorTaps\\_Two](#),  
    [SensorTaps\\_Three](#),  
    [SensorTaps\\_Four](#),  
    [SensorTaps\\_Eight](#),  
    [SensorTaps\\_Ten](#),  
    [NUM\\_SENSORTAPS](#) }
- enum [SensorDigitizationTapsEnums](#) {  
    [SensorDigitizationTaps\\_One](#),  
    [SensorDigitizationTaps\\_Two](#),  
    [SensorDigitizationTaps\\_Three](#),  
    [SensorDigitizationTaps\\_Four](#),  
    [SensorDigitizationTaps\\_Eight](#),  
    [SensorDigitizationTaps\\_Ten](#),  
    [NUM\\_SENSORDIGITIZATIONTAPS](#) }
- enum [RegionSelectorEnums](#) {  
    [RegionSelector\\_Region0](#),  
    [RegionSelector\\_Region1](#),  
    [RegionSelector\\_Region2](#),  
    [RegionSelector\\_All](#),  
    [NUM\\_REGIONSELECTOR](#) }
- enum [RegionModeEnums](#) {  
    [RegionMode\\_Off](#),  
    [RegionMode\\_On](#),  
    [NUM\\_REGIONMODE](#) }
- enum [RegionDestinationEnums](#) {  
    [RegionDestination\\_Stream0](#),  
    [RegionDestination\\_Stream1](#),  
    [RegionDestination\\_Stream2](#),  
    [NUM\\_REGIONDESTINATION](#) }
- enum [ImageComponentSelectorEnums](#) {



```
ImageComponentSelector_Intensity,  
ImageComponentSelector_Color,  
ImageComponentSelector_Infrared,  
ImageComponentSelector_Ultraviolet,  
ImageComponentSelector_Range,  
ImageComponentSelector_Disparity,  
ImageComponentSelector_Confidence,  
ImageComponentSelector_Scatter,  
NUM_IMAGECOMPONENTSELECTOR }
```

- enum [PixelFormatInfoSelectorEnums](#) {

PixelFormatInfoSelector BGra12p,

- ```
NUM_PIXELFORMATINFOSELECTOR }
```
- enum DeinterlacingEnums {  
Deinterlacing\_Off,  
Deinterlacing\_LineDuplication,  
Deinterlacing\_Weave,  
NUM\_DEINTERLACING }
  - enum ImageCompressionRateOptionEnums {  
ImageCompressionRateOption\_FixBitrate,  
ImageCompressionRateOption\_FixQuality,  
NUM\_IMAGECOMPRESSIONRATEOPTION }
  - enum ImageCompressionJPEGFormatOptionEnums {  
ImageCompressionJPEGFormatOption\_Lossless,  
ImageCompressionJPEGFormatOption\_BaselineStandard,  
ImageCompressionJPEGFormatOption\_BaselineOptimized,  
ImageCompressionJPEGFormatOption\_Progressive,  
NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION }
  - enum AcquisitionStatusSelectorEnums {  
AcquisitionStatusSelector\_AcquisitionTriggerWait,  
AcquisitionStatusSelector\_AcquisitionActive,  
AcquisitionStatusSelector\_AcquisitionTransfer,  
AcquisitionStatusSelector\_FrameTriggerWait,  
AcquisitionStatusSelector\_FrameActive,  
AcquisitionStatusSelector\_ExposureActive,  
NUM\_ACQUISITIONSTATUSSELECTOR }
  - enum ExposureTimeModeEnums {  
ExposureTimeMode\_Common,  
ExposureTimeMode\_Individual,  
NUM\_EXPOSURETIMEMODE }
  - enum ExposureTimeSelectorEnums {  
ExposureTimeSelector\_Common,  
ExposureTimeSelector\_Red,  
ExposureTimeSelector\_Green,  
ExposureTimeSelector\_Blue,  
ExposureTimeSelector\_Cyan,  
ExposureTimeSelector\_Magenta,  
ExposureTimeSelector\_Yellow,  
ExposureTimeSelector\_Infrared,  
ExposureTimeSelector\_Ultraviolet,  
ExposureTimeSelector\_Stage1,  
ExposureTimeSelector\_Stage2,  
NUM\_EXPOSURETIMESELECTOR }
  - enum GainAutoBalanceEnums {  
GainAutoBalance\_Off,  
GainAutoBalance\_Once,  
GainAutoBalance\_Continuous,  
NUM\_GAINAUTOBALANCE }
  - enum BlackLevelAutoEnums {  
BlackLevelAuto\_Off,  
BlackLevelAuto\_Once,  
BlackLevelAuto\_Continuous,  
NUM\_BLACKLEVELAUTO }
  - enum BlackLevelAutoBalanceEnums {  
BlackLevelAutoBalance\_Off,  
BlackLevelAutoBalance\_Once,  
BlackLevelAutoBalance\_Continuous,  
NUM\_BLACKLEVELAUTOBALANCE }
  - enum WhiteClipSelectorEnums {

```
WhiteClipSelector_All,  
WhiteClipSelector_Red,  
WhiteClipSelector_Green,  
WhiteClipSelector_Blue,  
WhiteClipSelector_Y,  
WhiteClipSelector_U,  
WhiteClipSelector_V,  
WhiteClipSelector_Tap1,  
WhiteClipSelector_Tap2,  
NUM_WHITECLIPSELECTOR }
```

- enum `TimerSelectorEnums` {  
    `TimerSelector_Timer0`,  
    `TimerSelector_Timer1`,  
    `TimerSelector_Timer2`,  
    `NUM_TIMERSELECTOR` }

- enum `TimerStatusEnums` {  
    `TimerStatus_TimerIdle`,  
    `TimerStatus_TimerTriggerWait`,  
    `TimerStatus_TimerActive`,  
    `TimerStatus_TimerCompleted`,  
    `NUM_TIMERSTATUS` }

- enum `TimerTriggerSourceEnums` {

```

TimerTriggerSource_Off,
TimerTriggerSource_AcquisitionTrigger,
TimerTriggerSource_AcquisitionStart,
TimerTriggerSource_AcquisitionEnd,
TimerTriggerSource_FrameTrigger,
TimerTriggerSource_FrameStart,
TimerTriggerSource_FrameEnd,
TimerTriggerSource_FrameBurstStart,
TimerTriggerSource_FrameBurstEnd,
TimerTriggerSource_LineTrigger,
TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

• enum TimerTriggerActivationEnums {
    TimerTriggerActivation_RisingEdge,
    TimerTriggerActivation_FallingEdge,
    TimerTriggerActivation_AnyEdge,
    TimerTriggerActivation_LevelHigh,
    TimerTriggerActivation_LevelLow,
    NUM_TIMERTRIGGERACTIVATION }

• enum EncoderSelectorEnums {
    EncoderSelector_Encoder0,
    EncoderSelector_Encoder1,
    EncoderSelector_Encoder2,
    NUM_ENCODERSELECTOR }

• enum EncoderSourceAEnums {

```

```
EncoderSourceA_Off,  
EncoderSourceA_Line0,  
EncoderSourceA_Line1,  
EncoderSourceA_Line2,  
NUM_ENCODERSOURCEA }
```

- `enum EncoderSourceBEnums {`  
EncoderSourceB\_Off,  
EncoderSourceB\_Line0,  
EncoderSourceB\_Line1,  
EncoderSourceB\_Line2,  
NUM\_ENCODERSOURCEB }

- `enum EncoderModeEnums {`  
EncoderMode\_FourPhase,  
EncoderMode\_HighResolution,  
NUM\_ENCODERMODE }

- `enum EncoderOutputModeEnums {`  
EncoderOutputMode\_Off,  
EncoderOutputMode\_PositionUp,  
EncoderOutputMode\_PositionDown,  
EncoderOutputMode\_DirectionUp,  
EncoderOutputMode\_DirectionDown,  
EncoderOutputMode\_Motion,  
NUM\_ENCODEROUTPUTMODE }

- `enum EncoderStatusEnums {`  
EncoderStatus\_EncoderUp,  
EncoderStatus\_EncoderDown,  
EncoderStatus\_EncoderIdle,  
EncoderStatus\_EncoderStatic,  
NUM\_ENCODERSTATUS }

- `enum EncoderResetSourceEnums {`

```

EncoderResetSource_Off,
EncoderResetSource_AcquisitionTrigger,
EncoderResetSource_AcquisitionStart,
EncoderResetSource_AcquisitionEnd,
EncoderResetSource_FrameTrigger,
EncoderResetSource_FrameStart,
EncoderResetSource_FrameEnd,
EncoderResetSource_ExposureStart,
EncoderResetSource_ExposureEnd,
EncoderResetSource_Line0,
EncoderResetSource_Line1,
EncoderResetSource_Line2,
EncoderResetSource_Counter0Start,
EncoderResetSource_Counter1Start,
EncoderResetSource_Counter2Start,
EncoderResetSource_Counter0End,
EncoderResetSource_Counter1End,
EncoderResetSource_Counter2End,
EncoderResetSource_Timer0Start,
EncoderResetSource_Timer1Start,
EncoderResetSource_Timer2Start,
EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,
EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }

• enum EncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }

• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }

• enum ActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }

• enum SourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,

```

```

NUM_SOURCESELECTOR }
• enum TransferSelectorEnums {
    TransferSelector_Stream0,
    TransferSelector_Stream1,
    TransferSelector_Stream2,
    TransferSelector_All,
    NUM_TRANSFERSELECTOR }
• enum TransferTriggerSelectorEnums {
    TransferTriggerSelector_TransferStart,
    TransferTriggerSelector_TransferStop,
    TransferTriggerSelector_TransferAbort,
    TransferTriggerSelector_TransferPause,
    TransferTriggerSelector_TransferResume,
    TransferTriggerSelector_TransferActive,
    TransferTriggerSelector_TransferBurstStart,
    TransferTriggerSelector_TransferBurstStop,
    NUM_TRANSFERTRIGGERSELECTOR }
• enum TransferTriggerModeEnums {
    TransferTriggerMode_Off,
    TransferTriggerMode_On,
    NUM_TRANSFERTRIGGERMODE }
• enum TransferTriggerSourceEnums {
    TransferTriggerSource_Line0,
    TransferTriggerSource_Line1,
    TransferTriggerSource_Line2,
    TransferTriggerSource_Counter0Start,
    TransferTriggerSource_Counter1Start,
    TransferTriggerSource_Counter2Start,
    TransferTriggerSource_Counter0End,
    TransferTriggerSource_Counter1End,
    TransferTriggerSource_Counter2End,
    TransferTriggerSource_Timer0Start,
    TransferTriggerSource_Timer1Start,
    TransferTriggerSource_Timer2Start,
    TransferTriggerSource_Timer0End,
    TransferTriggerSource_Timer1End,
    TransferTriggerSource_Timer2End,
    TransferTriggerSource_SoftwareSignal0,
    TransferTriggerSource_SoftwareSignal1,
    TransferTriggerSource_SoftwareSignal2,
    TransferTriggerSource_Action0,
    TransferTriggerSource_Action1,
    TransferTriggerSource_Action2,
    NUM_TRANSFERTRIGGERSOURCE }
• enum TransferTriggerActivationEnums {
    TransferTriggerActivation_RisingEdge,
    TransferTriggerActivation_FallingEdge,
    TransferTriggerActivation_AnyEdge,
    TransferTriggerActivation_LevelHigh,
    TransferTriggerActivation_LevelLow,
    NUM_TRANSFERTRIGGERACTIVATION }
• enum TransferStatusSelectorEnums {
    TransferStatusSelector_Streaming,
    TransferStatusSelector_Paused,
    TransferStatusSelector_Stopping,
    TransferStatusSelector_Stopped,
    TransferStatusSelector_QueueOverflow,
    NUM_TRANSFERSTATUSSELECTOR }

```



- enum TransferComponentSelectorEnums {  
TransferComponentSelector\_Red,  
TransferComponentSelector\_Green,  
TransferComponentSelector\_Blue,  
TransferComponentSelector\_All,  
NUM\_TRANSFERCOMPONENTSELECTOR }
- enum Scan3dDistanceUnitEnums {  
Scan3dDistanceUnit\_Millimeter,  
Scan3dDistanceUnit\_Inch,  
NUM\_SCAN3DDISTANCEUNIT }
- enum Scan3dCoordinateSystemEnums {  
Scan3dCoordinateSystem\_Cartesian,  
Scan3dCoordinateSystem\_Spherical,  
Scan3dCoordinateSystem\_Cylindrical,  
NUM\_SCAN3DCOORDINATESYSTEM }
- enum Scan3dOutputModeEnums {  
Scan3dOutputMode\_UncalibratedC,  
Scan3dOutputMode\_CalibratedABC\_Grid,  
Scan3dOutputMode\_CalibratedABC\_PointCloud,  
Scan3dOutputMode\_CalibratedAC,  
Scan3dOutputMode\_CalibratedAC\_Linescan,  
Scan3dOutputMode\_CalibratedC,  
Scan3dOutputMode\_CalibratedC\_Linescan,  
Scan3dOutputMode\_RectifiedC,  
Scan3dOutputMode\_RectifiedC\_Linescan,  
Scan3dOutputMode\_DisparityC,  
Scan3dOutputMode\_DisparityC\_Linescan,  
NUM\_SCAN3DOUTPUTMODE }
- enum Scan3dCoordinateSystemReferenceEnums {  
Scan3dCoordinateSystemReference\_Anchor,  
Scan3dCoordinateSystemReference\_Transformed,  
NUM\_SCAN3DCOORDINATESYSTEMREFERENCE }
- enum Scan3dCoordinateSelectorEnums {  
Scan3dCoordinateSelector\_CoordinateA,  
Scan3dCoordinateSelector\_CoordinateB,  
Scan3dCoordinateSelector\_CoordinateC,  
NUM\_SCAN3DCOORDINATESELECTOR }
- enum Scan3dCoordinateTransformSelectorEnums {  
Scan3dCoordinateTransformSelector\_RotationX,  
Scan3dCoordinateTransformSelector\_RotationY,  
Scan3dCoordinateTransformSelector\_RotationZ,  
Scan3dCoordinateTransformSelector\_TranslationX,  
Scan3dCoordinateTransformSelector\_TranslationY,  
Scan3dCoordinateTransformSelector\_TranslationZ,  
NUM\_SCAN3DCOORDINATETRANSFORMSELECTOR }
- enum Scan3dCoordinateReferenceSelectorEnums {  
Scan3dCoordinateReferenceSelector\_RotationX,  
Scan3dCoordinateReferenceSelector\_RotationY,  
Scan3dCoordinateReferenceSelector\_RotationZ,  
Scan3dCoordinateReferenceSelector\_TranslationX,  
Scan3dCoordinateReferenceSelector\_TranslationY,  
Scan3dCoordinateReferenceSelector\_TranslationZ,  
NUM\_SCAN3DCOORDINATEREFERENCESELECTOR }
- enum ChunkImageComponentEnums {

```

    ChunkImageComponent_Intensity,
    ChunkImageComponent_Color,
    ChunkImageComponent_Infrared,
    ChunkImageComponent_Ultraviolet,
    ChunkImageComponent_Range,
    ChunkImageComponent_Disparity,
    ChunkImageComponent_Confidence,
    ChunkImageComponent_Scatter,
    NUM_CHUNKIMAGECOMPONENT }
• enum ChunkCounterSelectorEnums {
    ChunkCounterSelector_Counter0,
    ChunkCounterSelector_Counter1,
    ChunkCounterSelector_Counter2,
    NUM_CHUNKCOUNTERSELECTOR }
• enum ChunkTimerSelectorEnums {
    ChunkTimerSelector_Timer0,
    ChunkTimerSelector_Timer1,
    ChunkTimerSelector_Timer2,
    NUM_CHUNKTIMERSELECTOR }
• enum ChunkEncoderSelectorEnums {
    ChunkEncoderSelector_Encoder0,
    ChunkEncoderSelector_Encoder1,
    ChunkEncoderSelector_Encoder2,
    NUM_CHUNKENCODERSELECTOR }
• enum ChunkEncoderStatusEnums {
    ChunkEncoderStatus_EncoderUp,
    ChunkEncoderStatus_EncoderDown,
    ChunkEncoderStatus_EncoderIdle,
    ChunkEncoderStatus_EncoderStatic,
    NUM_CHUNKENCODERSTATUS }
• enum ChunkExposureTimeSelectorEnums {
    ChunkExposureTimeSelector_Common,
    ChunkExposureTimeSelector_Red,
    ChunkExposureTimeSelector_Green,
    ChunkExposureTimeSelector_Blue,
    ChunkExposureTimeSelector_Cyan,
    ChunkExposureTimeSelector_Magenta,
    ChunkExposureTimeSelector_Yellow,
    ChunkExposureTimeSelector_Infrared,
    ChunkExposureTimeSelector_Ultraviolet,
    ChunkExposureTimeSelector_Stage1,
    ChunkExposureTimeSelector_Stage2,
    NUM_CHUNKEXPOSURETIMESELECTOR }
• enum ChunkSourceIDEnums {
    ChunkSourceID_Source0,
    ChunkSourceID_Source1,
    ChunkSourceID_Source2,
    NUM_CHUNKSOURCEID }
• enum ChunkRegionIDEnums {
    ChunkRegionID_Region0,
    ChunkRegionID_Region1,
    ChunkRegionID_Region2,
    NUM_CHUNKREGIONID }
• enum ChunkTransferStreamIDEnums {
    ChunkTransferStreamID_Stream0,
    ChunkTransferStreamID_Stream1,
    ChunkTransferStreamID_Stream2,
    ChunkTransferStreamID_Stream3,

```

- ```
NUM_CHUNKTRANSFERSTREAMID }
```
- enum ChunkScan3dDistanceUnitEnums {  
 ChunkScan3dDistanceUnit\_Millimeter,  
 ChunkScan3dDistanceUnit\_Inch,  
 NUM\_CHUNKSCAN3DDISTANCEUNIT }
  - enum ChunkScan3dOutputModeEnums {  
 ChunkScan3dOutputMode\_UncalibratedC,  
 ChunkScan3dOutputMode\_CalibratedABC\_Grid,  
 ChunkScan3dOutputMode\_CalibratedABC\_PointCloud,  
 ChunkScan3dOutputMode\_CalibratedAC,  
 ChunkScan3dOutputMode\_CalibratedAC\_Linescan,  
 ChunkScan3dOutputMode\_CalibratedC,  
 ChunkScan3dOutputMode\_CalibratedC\_Linescan,  
 ChunkScan3dOutputMode\_RectifiedC,  
 ChunkScan3dOutputMode\_RectifiedC\_Linescan,  
 ChunkScan3dOutputMode\_DisparityC,  
 ChunkScan3dOutputMode\_DisparityC\_Linescan,  
 NUM\_CHUNKSCAN3DOUTPUTMODE }
  - enum ChunkScan3dCoordinateSystemEnums {  
 ChunkScan3dCoordinateSystem\_Cartesian,  
 ChunkScan3dCoordinateSystem\_Spherical,  
 ChunkScan3dCoordinateSystem\_Cylindrical,  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEM }
  - enum ChunkScan3dCoordinateSystemReferenceEnums {  
 ChunkScan3dCoordinateSystemReference\_Anchor,  
 ChunkScan3dCoordinateSystemReference\_Transformed,  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }
  - enum ChunkScan3dCoordinateSelectorEnums {  
 ChunkScan3dCoordinateSelector\_CoordinateA,  
 ChunkScan3dCoordinateSelector\_CoordinateB,  
 ChunkScan3dCoordinateSelector\_CoordinateC,  
 NUM\_CHUNKSCAN3DCOORDINATESELECTOR }
  - enum ChunkScan3dCoordinateTransformSelectorEnums {  
 ChunkScan3dCoordinateTransformSelector\_RotationX,  
 ChunkScan3dCoordinateTransformSelector\_RotationY,  
 ChunkScan3dCoordinateTransformSelector\_RotationZ,  
 ChunkScan3dCoordinateTransformSelector\_TranslationX,  
 ChunkScan3dCoordinateTransformSelector\_TranslationY,  
 ChunkScan3dCoordinateTransformSelector\_TranslationZ,  
 NUM\_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }
  - enum ChunkScan3dCoordinateReferenceSelectorEnums {  
 ChunkScan3dCoordinateReferenceSelector\_RotationX,  
 ChunkScan3dCoordinateReferenceSelector\_RotationY,  
 ChunkScan3dCoordinateReferenceSelector\_RotationZ,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationX,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationY,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationZ,  
 NUM\_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }
  - enum DeviceTapGeometryEnums {

```

DeviceTapGeometry_Geometry_1X_1Y,
DeviceTapGeometry_Geometry_1X2_1Y,
DeviceTapGeometry_Geometry_1X2_1Y2,
DeviceTapGeometry_Geometry_2X_1Y,
DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
DeviceTapGeometry_Geometry_2XE_1Y2,
DeviceTapGeometry_Geometry_2XM_1Y,
DeviceTapGeometry_Geometry_2XM_1Y2,
DeviceTapGeometry_Geometry_1X_1Y2,
DeviceTapGeometry_Geometry_1X_2YE,
DeviceTapGeometry_Geometry_1X3_1Y,
DeviceTapGeometry_Geometry_3X_1Y,
DeviceTapGeometry_Geometry_1X,
DeviceTapGeometry_Geometry_1X2,
DeviceTapGeometry_Geometry_2X,
DeviceTapGeometry_Geometry_2XE,
DeviceTapGeometry_Geometry_2XM,
DeviceTapGeometry_Geometry_1X3,
DeviceTapGeometry_Geometry_3X,
DeviceTapGeometry_Geometry_1X4_1Y,
DeviceTapGeometry_Geometry_4X_1Y,
DeviceTapGeometry_Geometry_2X2_1Y,
DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
DeviceTapGeometry_Geometry_1X2_2YE,
DeviceTapGeometry_Geometry_2X_2YE,
DeviceTapGeometry_Geometry_2XE_2YE,
DeviceTapGeometry_Geometry_2XM_2YE,
DeviceTapGeometry_Geometry_1X4,
DeviceTapGeometry_Geometry_4X,
DeviceTapGeometry_Geometry_2X2,
DeviceTapGeometry_Geometry_2X2E,
DeviceTapGeometry_Geometry_2X2M,
DeviceTapGeometry_Geometry_1X8_1Y,
DeviceTapGeometry_Geometry_8X_1Y,
DeviceTapGeometry_Geometry_4X2_1Y,
DeviceTapGeometry_Geometry_2X2E_2YE,
DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

• enum GevPhysicalLinkConfigurationEnums {
    GevPhysicalLinkConfiguration_SingleLink,
    GevPhysicalLinkConfiguration_MultiLink,
    GevPhysicalLinkConfiguration_StaticLAG,
    GevPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVPHYSICALLINKCONFIGURATION }

• enum GevCurrentPhysicalLinkConfigurationEnums {
    GevCurrentPhysicalLinkConfiguration_SingleLink,
    GevCurrentPhysicalLinkConfiguration_MultiLink,
    GevCurrentPhysicalLinkConfiguration_StaticLAG,
    GevCurrentPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }

```

- enum `GevIPConfigurationStatusEnums` {  
    `GevIPConfigurationStatus_None`,  
    `GevIPConfigurationStatus_PersistentIP`,  
    `GevIPConfigurationStatus_DHCP`,  
    `GevIPConfigurationStatus_LLA`,  
    `GevIPConfigurationStatus_ForceIP`,  
    `NUM_GEVIPCONFIGURATIONSTATUS` }
  
- enum `GevGVCPExtendedStatusCodesSelectorEnums` {  
    `GevGVCPExtendedStatusCodesSelector_Version1_1`,  
    `GevGVCPExtendedStatusCodesSelector_Version2_0`,  
    `NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR` }
  
- enum `GevGVSPExtendedIDModeEnums` {  
    `GevGVSPExtendedIDMode_Off`,  
    `GevGVSPExtendedIDMode_On`,  
    `NUM_GEVGVSPEXTENDEDIDMODE` }
  
- enum `CIConfigurationEnums` {  
    `CIConfiguration_Base`,  
    `CIConfiguration_Medium`,  
    `CIConfiguration_Full`,  
    `CIConfiguration_DualBase`,  
    `CIConfiguration_EightyBit`,  
    `NUM_CLCONFIGURATION` }
  
- enum `CITimeSlotsCountEnums` {  
    `CITimeSlotsCount_One`,  
    `CITimeSlotsCount_Two`,  
    `CITimeSlotsCount_Three`,  
    `NUM_CLTIMESLOTSCOUNT` }
  
- enum `CxpLinkConfigurationStatusEnums` {

```
CxpLinkConfigurationStatus_None,  
CxpLinkConfigurationStatus_Pending,  
CxpLinkConfigurationStatus_CXP1_X1,  
CxpLinkConfigurationStatus_CXP2_X1,  
CxpLinkConfigurationStatus_CXP3_X1,  
CxpLinkConfigurationStatus_CXP5_X1,  
CxpLinkConfigurationStatus_CXP6_X1,  
CxpLinkConfigurationStatus_CXP1_X2,  
CxpLinkConfigurationStatus_CXP2_X2,  
CxpLinkConfigurationStatus_CXP3_X2,  
CxpLinkConfigurationStatus_CXP5_X2,  
CxpLinkConfigurationStatus_CXP6_X2,  
CxpLinkConfigurationStatus_CXP1_X3,  
CxpLinkConfigurationStatus_CXP2_X3,  
CxpLinkConfigurationStatus_CXP3_X3,  
CxpLinkConfigurationStatus_CXP5_X3,  
CxpLinkConfigurationStatus_CXP6_X3,  
CxpLinkConfigurationStatus_CXP1_X4,  
CxpLinkConfigurationStatus_CXP2_X4,  
CxpLinkConfigurationStatus_CXP3_X4,  
CxpLinkConfigurationStatus_CXP5_X4,  
CxpLinkConfigurationStatus_CXP6_X4,  
CxpLinkConfigurationStatus_CXP1_X5,  
CxpLinkConfigurationStatus_CXP2_X5,  
CxpLinkConfigurationStatus_CXP3_X5,  
CxpLinkConfigurationStatus_CXP5_X5,  
CxpLinkConfigurationStatus_CXP6_X5,  
CxpLinkConfigurationStatus_CXP1_X6,  
CxpLinkConfigurationStatus_CXP2_X6,  
CxpLinkConfigurationStatus_CXP3_X6,  
CxpLinkConfigurationStatus_CXP5_X6,  
CxpLinkConfigurationStatus_CXP6_X6,  
NUM_CXPLINKCONFIGURATIONSTATUS }
```

• enum [CxpLinkConfigurationPreferredEnums](#) {

```
CxpLinkConfigurationPreferred_CXP1_X1,  
CxpLinkConfigurationPreferred_CXP2_X1,  
CxpLinkConfigurationPreferred_CXP3_X1,  
CxpLinkConfigurationPreferred_CXP5_X1,  
CxpLinkConfigurationPreferred_CXP6_X1,  
CxpLinkConfigurationPreferred_CXP1_X2,  
CxpLinkConfigurationPreferred_CXP2_X2,  
CxpLinkConfigurationPreferred_CXP3_X2,  
CxpLinkConfigurationPreferred_CXP5_X2,  
CxpLinkConfigurationPreferred_CXP6_X2,  
CxpLinkConfigurationPreferred_CXP1_X3,  
CxpLinkConfigurationPreferred_CXP2_X3,  
CxpLinkConfigurationPreferred_CXP3_X3,  
CxpLinkConfigurationPreferred_CXP5_X3,  
CxpLinkConfigurationPreferred_CXP6_X3,  
CxpLinkConfigurationPreferred_CXP1_X4,  
CxpLinkConfigurationPreferred_CXP2_X4,  
CxpLinkConfigurationPreferred_CXP3_X4,  
CxpLinkConfigurationPreferred_CXP5_X4,  
CxpLinkConfigurationPreferred_CXP6_X4,  
CxpLinkConfigurationPreferred_CXP1_X5,  
CxpLinkConfigurationPreferred_CXP2_X5,  
CxpLinkConfigurationPreferred_CXP3_X5,  
CxpLinkConfigurationPreferred_CXP5_X5,  
CxpLinkConfigurationPreferred_CXP6_X5,  
CxpLinkConfigurationPreferred_CXP1_X6,  
CxpLinkConfigurationPreferred_CXP2_X6,  
CxpLinkConfigurationPreferred_CXP3_X6,  
CxpLinkConfigurationPreferred_CXP5_X6,  
CxpLinkConfigurationPreferred_CXP6_X6,  
NUM_CXPLINKCONFIGURATIONPREFERRED }
```

- enum [CxpLinkConfigurationEnums](#) {

```

CxpLinkConfiguration_Auto,
CxpLinkConfiguration_CXP1_X1,
CxpLinkConfiguration_CXP2_X1,
CxpLinkConfiguration_CXP3_X1,
CxpLinkConfiguration_CXP5_X1,
CxpLinkConfiguration_CXP6_X1,
CxpLinkConfiguration_CXP1_X2,
CxpLinkConfiguration_CXP2_X2,
CxpLinkConfiguration_CXP3_X2,
CxpLinkConfiguration_CXP5_X2,
CxpLinkConfiguration_CXP6_X2,
CxpLinkConfiguration_CXP1_X3,
CxpLinkConfiguration_CXP2_X3,
CxpLinkConfiguration_CXP3_X3,
CxpLinkConfiguration_CXP5_X3,
CxpLinkConfiguration_CXP6_X3,
CxpLinkConfiguration_CXP1_X4,
CxpLinkConfiguration_CXP2_X4,
CxpLinkConfiguration_CXP3_X4,
CxpLinkConfiguration_CXP5_X4,
CxpLinkConfiguration_CXP6_X4,
CxpLinkConfiguration_CXP1_X5,
CxpLinkConfiguration_CXP2_X5,
CxpLinkConfiguration_CXP3_X5,
CxpLinkConfiguration_CXP5_X5,
CxpLinkConfiguration_CXP6_X5,
CxpLinkConfiguration_CXP1_X6,
CxpLinkConfiguration_CXP2_X6,
CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }
• enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }
• enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPCXPSTATUS }

```

### 8.8.1 Detailed Description

### 8.8.2 Enumeration Type Documentation

#### 8.8.2.1 enum AcquisitionModeEnums

< Sets the acquisition mode of the device. Continuous: acquires images continuously. Multi Frame: acquires a specified number of images before stopping acquisition. Single Frame: acquires 1 image before stopping acquisition.

Enumerator

```

AcquisitionMode_Continuous
AcquisitionMode_SingleFrame
AcquisitionMode_MultiFrame
NUM_ACQUISITIONMODE

```



## 8.8.2.2 enum AcquisitionStatusSelectorEnums

< Selects the internal acquisition signal to read using AcquisitionStatus.

Enumerator

**AcquisitionStatusSelector\_AcquisitionTriggerWait** Device is currently waiting for a trigger for the capture of one or many frames.

**AcquisitionStatusSelector\_AcquisitionActive** Device is currently doing an acquisition of one or many frames.

**AcquisitionStatusSelector\_AcquisitionTransfer** Device is currently transferring an acquisition of one or many frames.

**AcquisitionStatusSelector\_FrameTriggerWait** Device is currently waiting for a frame start trigger.

**AcquisitionStatusSelector\_FrameActive** Device is currently doing the capture of a frame.

**AcquisitionStatusSelector\_ExposureActive** Device is doing the exposure of a frame.

**NUM\_ACQUISITIONSTATUSSELECTOR**

## 8.8.2.3 enum ActionUnconditionalModeEnums

< Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

Enumerator

**ActionUnconditionalMode\_Off** Unconditional mode is disabled.

**ActionUnconditionalMode\_On** Unconditional mode is enabled.

**NUM\_ACTIONUNCONDITIONALMODE**

## 8.8.2.4 enum AdcBitDepthEnums

< Selects which ADC bit depth to use. A higher ADC bit depth results in better image quality but slower maximum frame rate.

Enumerator

**AdcBitDepth\_Bit8**

**AdcBitDepth\_Bit10**

**AdcBitDepth\_Bit12**

**AdcBitDepth\_Bit14**

**NUM\_ADCBITDEPTH**

## 8.8.2.5 enum AutoAlgorithmSelectorEnums

< Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.

Enumerator

**AutoAlgorithmSelector\_Awb** Selects the Auto White Balance algorithm.

**AutoAlgorithmSelector\_Ae** Selects the Auto Exposure algorithm.

**NUM\_AUTOALGORITHMSELECTOR**

### 8.8.2.6 enum AutoExposureControlPriorityEnums

< Selects whether to adjust gain or exposure first. When gain priority is selected, the camera fixes the gain to 0 dB, and the exposure is adjusted according to the target grey level. If the maximum exposure is reached before the target grey level is hit, the gain starts to change to meet the target. This mode is used to have the minimum noise. When exposure priority is selected, the camera sets the exposure to a small value (default is 5 ms). The gain is adjusted according to the target grey level. If maximum gain is reached before the target grey level is hit, the exposure starts to change to meet the target. This mode is used to capture fast motion.

Enumerator

***AutoExposureControlPriority\_Gain***  
***AutoExposureControlPriority\_ExposureTime***  
***NUM\_AUTOEXPOSURECONTROLPRIORITY***

### 8.8.2.7 enum AutoExposureLightingModeEnums

< Selects a lighting mode: Backlight, Frontlight or Normal (default). a. Backlight compensation: used when a strong light is coming from the back of the object. b. Frontlight compensation: used when a strong light is shining in the front of the object while the background is dark. c. Normal lighting: used when the object is not under backlight or frontlight conditions. When normal lighting is selected, metering modes are available.

Enumerator

***AutoExposureLightingMode\_AutoDetect***  
***AutoExposureLightingMode\_Backlight***  
***AutoExposureLightingMode\_Frontlight***  
***AutoExposureLightingMode\_Normal***  
***NUM\_AUTOEXPOSURELIGHTINGMODE***

### 8.8.2.8 enum AutoExposureMeteringModeEnums

< Selects a metering mode: average, spot, or partial metering. a. Average: Measures the light from the entire scene uniformly to determine the final exposure value. Every portion of the exposed area has the same contribution. b. Spot: Measures a small area (about 3%) in the center of the scene while the rest of the scene is ignored. This mode is used when the scene has a high contrast and the object of interest is relatively small. c. Partial: Measures the light from a larger area (about 11%) in the center of the scene. This mode is used when very dark or bright regions appear at the edge of the frame. Note: Metering mode is available only when Lighting Mode Selector is Normal.

Enumerator

***AutoExposureMeteringMode\_Average***  
***AutoExposureMeteringMode\_Spot***  
***AutoExposureMeteringMode\_Partial***  
***AutoExposureMeteringMode\_CenterWeighted***  
***AutoExposureMeteringMode\_HistogramPeak***  
***NUM\_AUTOEXPOSUREMETERINGMODE***

## 8.8.2.9 enum AutoExposureTargetGreyValueAutoEnums

< This indicates whether the target image grey level is automatically set by the camera or manually set by the user. Note that the target grey level is in the linear domain before gamma correction is applied.

## Enumerator

**AutoExposureTargetGreyValueAuto\_Off** Target grey value is manually controlled

**AutoExposureTargetGreyValueAuto\_Continuous** Target grey value is constantly adapted by the device to maximize the dynamic range.

**NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO**

## 8.8.2.10 enum BalanceRatioSelectorEnums

< Selects a balance ratio to configure once a balance ratio control has been selected.

## Enumerator

**BalanceRatioSelector\_Red** Selects the red balance ratio control for adjustment. The red balance ratio is relative to the green channel.

**BalanceRatioSelector\_Blue** Selects the blue balance ratio control for adjustment. The blue balance ratio is relative to the green channel.

**NUM\_BALANCERATIOSELECTOR**

## 8.8.2.11 enum BalanceWhiteAutoEnums

< White Balance compensates for color shifts caused by different lighting conditions. It can be automatically or manually controlled. For manual control, set to Off. For automatic control, set to Once or Continuous.

## Enumerator

**BalanceWhiteAuto\_Off** Sets operation mode to Off, which is manual control.

**BalanceWhiteAuto\_Once** Sets operation mode to once. Once runs for a number of iterations and then sets White Balance Auto to Off.

**BalanceWhiteAuto\_Continuous** Sets operation mode to continuous. Continuous automatically adjusts values if the colors are imbalanced.

**NUM\_BALANCEWHITEAUTO**

## 8.8.2.12 enum BalanceWhiteAutoProfileEnums

< Selects the profile used by BalanceWhiteAuto.

## Enumerator

**BalanceWhiteAutoProfile\_Indoor** Indoor auto white balance Profile. Can be used to compensate for artificial lighting.

**BalanceWhiteAutoProfile\_Outdoor** Outdoor auto white balance profile. Designed for scenes with natural lighting.

**NUM\_BALANCEWHITEAUTOPROFILE**

## 8.8.2.13 enum BinningHorizontalModeEnums

&lt;

## Enumerator

**BinningHorizontalMode\_Sum** The response from the combined horizontal cells is added, resulting in increased sensitivity (a brighter image).

**BinningHorizontalMode\_Average** The response from the combined horizontal cells is averaged, resulting in increased signal/noise ratio. Not all sensors support average binning.

**NUM\_BINNINGHORIZONTALMODE**

## 8.8.2.14 enum BinningSelectorEnums

< Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.

## Enumerator

**BinningSelector\_All** The total amount of binning to be performed on the captured sensor data.

**BinningSelector\_Sensor** The portion of binning to be performed on the sensor directly.

**BinningSelector\_ISP** The portion of binning to be performed by the image signal processing engine (ISP) outside of the sensor. Note: the ISP can be disabled.

**NUM\_BINNINGSELECTOR**

## 8.8.2.15 enum BinningVerticalModeEnums

&lt;

## Enumerator

**BinningVerticalMode\_Sum** The response from the combined vertical cells is added, resulting in increased sensitivity (a brighter image).

**BinningVerticalMode\_Average** The response from the combined vertical cells is averaged, resulting in increased signal/noise ratio. Not all sensors support average binning.

**NUM\_BINNINGVERTICALMODE**

## 8.8.2.16 enum BlackLevelAutoBalanceEnums

< Controls the mode for automatic black level balancing between the sensor color channels or taps. The black level coefficients of each channel are adjusted so they are matched.

## Enumerator

**BlackLevelAutoBalance\_Off** Black level tap balancing is user controlled using BlackLevel.

**BlackLevelAutoBalance\_Once** Black level tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.

**BlackLevelAutoBalance\_Continuous** Black level tap balancing is constantly adjusted by the device.

**NUM\_BLACKLEVELAUTOBALANCE**

## 8.8.2.17 enum BlackLevelAutoEnums

< Controls the mode for automatic black level adjustment. The exact algorithm used to implement this adjustment is device-specific.

## Enumerator

***BlackLevelAuto\_Off*** Analog black level is user controlled using BlackLevel.

***BlackLevelAuto\_Once*** Analog black level is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.

***BlackLevelAuto\_Continuous*** Analog black level is constantly adjusted by the device.

***NUM\_BLACKLEVELAUTO***

## 8.8.2.18 enum BlackLevelSelectorEnums

< Selects which black level to control. Only All can be set by the user. Analog and Digital are read-only.

## Enumerator

***BlackLevelSelector\_All***

***BlackLevelSelector\_Analog***

***BlackLevelSelector\_Digital***

***NUM\_BLACKLEVELSELECTOR***

## 8.8.2.19 enum ChunkBlackLevelSelectorEnums

< Selects which black level to retrieve

## Enumerator

***ChunkBlackLevelSelector\_All***

***NUM\_CHUNKBLACKLEVELSELECTOR***

## 8.8.2.20 enum ChunkCounterSelectorEnums

< Selects which counter to retrieve data from.

## Enumerator

***ChunkCounterSelector\_Counter0*** Selects the counter 0.

***ChunkCounterSelector\_Counter1*** Selects the counter 1.

***ChunkCounterSelector\_Counter2*** Selects the counter 2.

***NUM\_CHUNKCOUNTERSELECTOR***

## 8.8.2.21 enum ChunkEncoderSelectorEnums

< Selects which Encoder to retrieve data from.

Enumerator

**ChunkEncoderSelector\_Encoder0** Selects the first Encoder.  
**ChunkEncoderSelector\_Encoder1** Selects the first Encoder.  
**ChunkEncoderSelector\_Encoder2** Selects the second Encoder.  
**NUM\_CHUNKENCODERSELECTOR**

## 8.8.2.22 enum ChunkEncoderStatusEnums

< Returns the motion status of the selected encoder.

Enumerator

**ChunkEncoderStatus\_EncoderUp** The encoder counter last incremented.  
**ChunkEncoderStatus\_EncoderDown** The encoder counter last decremented.  
**ChunkEncoderStatus\_EncoderIdle** The encoder is not active.  
**ChunkEncoderStatus\_EncoderStatic** No motion within the EncoderTimeout time.  
**NUM\_CHUNKENCODERSTATUS**

## 8.8.2.23 enum ChunkExposureTimeSelectorEnums

< Selects which exposure time is read by the ChunkExposureTime feature.

Enumerator

**ChunkExposureTimeSelector\_Common** Selects the common ExposureTime.  
**ChunkExposureTimeSelector\_Red** Selects the red common ExposureTime.  
**ChunkExposureTimeSelector\_Green** Selects the green ExposureTime.  
**ChunkExposureTimeSelector\_Blue** Selects the blue ExposureTime.  
**ChunkExposureTimeSelector\_Cyan** Selects the cyan common ExposureTime..  
**ChunkExposureTimeSelector\_Magenta** Selects the magenta ExposureTime..  
**ChunkExposureTimeSelector\_Yellow** Selects the yellow ExposureTime..  
**ChunkExposureTimeSelector\_Infrared** Selects the infrared ExposureTime.  
**ChunkExposureTimeSelector\_Ultraviolet** Selects the ultraviolet ExposureTime.  
**ChunkExposureTimeSelector\_Stage1** Selects the first stage ExposureTime.  
**ChunkExposureTimeSelector\_Stage2** Selects the second stage ExposureTime.  
**NUM\_CHUNKEXPOSURETIMESELECTOR**

## 8.8.2.24 enum ChunkGainSelectorEnums

< Selects which gain to retrieve

Enumerator

***ChunkGainSelector\_All***  
***ChunkGainSelector\_Red***  
***ChunkGainSelector\_Green***  
***ChunkGainSelector\_Blue***  
***NUM\_CHUNKGAINSELECTOR***

## 8.8.2.25 enum ChunkImageComponentEnums

< Returns the component of the payload image. This can be used to identify the image component of a generic part in a multipart transfer.

Enumerator

***ChunkImageComponent\_Intensity*** The image data is the intensity component.  
***ChunkImageComponent\_Color*** The image data is color component.  
***ChunkImageComponent\_Infrared*** The image data is infrared component.  
***ChunkImageComponent\_Ultraviolet*** The image data is the ultraviolet component.  
***ChunkImageComponent\_Range*** The image data is the range (distance) component.  
***ChunkImageComponent\_Disparity*** The image data is the disparity component.  
***ChunkImageComponent\_Confidence*** The image data is the confidence map component.  
***ChunkImageComponent\_Scatter*** The image data is the scatter component.  
***NUM\_CHUNKIMAGECOMPONENT***

## 8.8.2.26 enum ChunkPixelFormatEnums

< Format of the pixel provided by the camera

Enumerator

***ChunkPixelFormat\_Mono8***  
***ChunkPixelFormat\_Mono12Packed***  
***ChunkPixelFormat\_Mono16***  
***ChunkPixelFormat\_RGB8Packed***  
***ChunkPixelFormat\_YUV422Packed***  
***ChunkPixelFormat\_BayerGR8***  
***ChunkPixelFormat\_BayerRG8***  
***ChunkPixelFormat\_BayerGB8***  
***ChunkPixelFormat\_BayerBG8***  
***ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY***  
***NUM\_CHUNKPIXELFORMAT***

## 8.8.2.27 enum ChunkRegionIDEnums

< Returns the identifier of Region that the image comes from.

## Enumerator

**ChunkRegionID\_Region0** Image comes from the Region 0.

**ChunkRegionID\_Region1** Image comes from the Region 1.

**ChunkRegionID\_Region2** Image comes from the Region 2.

**NUM\_CHUNKREGIONID**

## 8.8.2.28 enum ChunkScan3dCoordinateReferenceSelectorEnums

< Selector to read a coordinate system reference value defining the transform of a point from one system to the other.

## Enumerator

**ChunkScan3dCoordinateReferenceSelector\_RotationX** Rotation around X axis.

**ChunkScan3dCoordinateReferenceSelector\_RotationY** Rotation around Y axis.

**ChunkScan3dCoordinateReferenceSelector\_RotationZ** Rotation around Z axis.

**ChunkScan3dCoordinateReferenceSelector\_TranslationX** X axis translation.

**ChunkScan3dCoordinateReferenceSelector\_TranslationY** Y axis translation.

**ChunkScan3dCoordinateReferenceSelector\_TranslationZ** Z axis translation.

**NUM\_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR**

## 8.8.2.29 enum ChunkScan3dCoordinateSelectorEnums

< Selects which Coordinate to retrieve data from.

## Enumerator

**ChunkScan3dCoordinateSelector\_CoordinateA** The first (X or Theta) coordinate

**ChunkScan3dCoordinateSelector\_CoordinateB** The second (Y or Phi) coordinate

**ChunkScan3dCoordinateSelector\_CoordinateC** The third (Z or Rho) coordinate.

**NUM\_CHUNKSCAN3DCOORDINATESELECTOR**

## 8.8.2.30 enum ChunkScan3dCoordinateSystemEnums

< Returns the Coordinate System of the image included in the payload.

## Enumerator

**ChunkScan3dCoordinateSystem\_Cartesian** Default value. 3-axis orthogonal, right-hand X-Y-Z.

**ChunkScan3dCoordinateSystem\_Spherical** A Theta-Phi-Rho coordinate system.

**ChunkScan3dCoordinateSystem\_Cylindrical** A Theta-Y-Rho coordinate system.

**NUM\_CHUNKSCAN3DCOORDINATESYSTEM**



## 8.8.2.31 enum ChunkScan3dCoordinateSystemReferenceEnums

< Returns the Coordinate [System](#) Position of the image included in the payload.

## Enumerator

**ChunkScan3dCoordinateSystemReference\_Anchor** Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.

**ChunkScan3dCoordinateSystemReference\_Transformed** Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices.

**NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE**

## 8.8.2.32 enum ChunkScan3dCoordinateTransformSelectorEnums

< Selector for transform values.

## Enumerator

**ChunkScan3dCoordinateTransformSelector\_RotationX** Rotation around X axis.

**ChunkScan3dCoordinateTransformSelector\_RotationY** Rotation around Y axis.

**ChunkScan3dCoordinateTransformSelector\_RotationZ** Rotation around Z axis.

**ChunkScan3dCoordinateTransformSelector\_TranslationX** Translation along X axis.

**ChunkScan3dCoordinateTransformSelector\_TranslationY** Translation along Y axis.

**ChunkScan3dCoordinateTransformSelector\_TranslationZ** Translation along Z axis.

**NUM\_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR**

## 8.8.2.33 enum ChunkScan3dDistanceUnitEnums

< Returns the Distance Unit of the payload image.

## Enumerator

**ChunkScan3dDistanceUnit\_Millimeter** Default value. Distance values are in millimeter units.

**ChunkScan3dDistanceUnit\_Inch** Distance values are in inch units.

**NUM\_CHUNKSCAN3DDISTANCEUNIT**

## 8.8.2.34 enum ChunkScan3dOutputModeEnums

< Returns the Calibrated Mode of the payload image.

## Enumerator

**ChunkScan3dOutputMode\_UncalibratedC** Uncalibrated 2.5D Depth map. The distance data does not represent a physical unit and may be non-linear. The data is a 2.5D range map only.

**ChunkScan3dOutputMode\_CalibratedABC\_Grid** 3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.

**ChunkScan3dOutputMode\_CalibratedABC\_PointCloud** 3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.

**ChunkScan3dOutputMode\_CalibratedAC** 2 Coordinates with fixed B sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis uses the scale and offset parameters for the B axis.

**ChunkScan3dOutputMode\_CalibratedAC\_Linescan** 2 Coordinates with varying sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis comes from the encoder chunk value.

**ChunkScan3dOutputMode\_CalibratedC** Calibrated 2.5D Depth map. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. No information on X-Y axes available.

**ChunkScan3dOutputMode\_CalibratedC\_Linescan** Depth Map with varying B sampling. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. The B (Y) axis comes from the encoder chunk value.

**ChunkScan3dOutputMode\_RectifiedC** Rectified 2.5D Depth map. The distance data has been rectified to a uniform sampling pattern in the X and Y direction. The data is a 2.5D range map only. If a complete 3D point cloud is rectified but transmitted as explicit coordinates it should be transmitted as one of the "CalibratedABC" formats.

**ChunkScan3dOutputMode\_RectifiedC\_Linescan** Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using Coord3D\_C pixels. The B (Y) axis comes from the encoder chunk value.

**ChunkScan3dOutputMode\_DisparityC** Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.

**ChunkScan3dOutputMode\_DisparityC\_Linescan** Disparity 2.5D Depth map with varying B sampling. The distance is inversely proportional to the pixel (disparity) value. The B (Y) axis comes from the encoder chunk value.

**NUM\_CHUNKSCAN3DOUTPUTMODE**

## 8.8.2.35 enum ChunkSelectorEnums

< Selects which chunk data to enable or disable.

## Enumerator

**ChunkSelector\_Image**

**ChunkSelector\_CRC**

**ChunkSelector\_FrameID**

**ChunkSelector\_OffsetX**

**ChunkSelector\_OffsetY**

**ChunkSelector\_Width**

**ChunkSelector\_Height**

***ChunkSelector\_ExposureTime***  
***ChunkSelector\_Gain***  
***ChunkSelector\_BlackLevel***  
***ChunkSelector\_PixelFormat***  
***ChunkSelector\_Timestamp***  
***ChunkSelector\_SequencerSetActive***  
***ChunkSelector\_SerialData***  
***ChunkSelector\_ExposureEndLineStatusAll***  
***NUM\_CHUNKSELECTOR***

#### 8.8.2.36 enum ChunkSourceIDEnums

< Returns the identifier of Source that the image comes from.

Enumerator

***ChunkSourceID\_Source0*** Image comes from the Source 0.  
***ChunkSourceID\_Source1*** Image comes from the Source 1.  
***ChunkSourceID\_Source2*** Image comes from the Source 2.  
***NUM\_CHUNKSOURCEID***

#### 8.8.2.37 enum ChunkTimerSelectorEnums

< Selects which Timer to retrieve data from.

Enumerator

***ChunkTimerSelector\_Timer0*** Selects the first Timer.  
***ChunkTimerSelector\_Timer1*** Selects the first Timer.  
***ChunkTimerSelector\_Timer2*** Selects the second Timer.  
***NUM\_CHUNKTIMERSELECTOR***

#### 8.8.2.38 enum ChunkTransferStreamIDEnums

< Returns identifier of the stream that generated this block.

Enumerator

***ChunkTransferStreamID\_Stream0*** Data comes from Stream0.  
***ChunkTransferStreamID\_Stream1*** Data comes from Stream1.  
***ChunkTransferStreamID\_Stream2*** Data comes from Stream2.  
***ChunkTransferStreamID\_Stream3*** Data comes from Stream3.  
***NUM\_CHUNKTRANSFERSTREAMID***

## 8.8.2.39 enum CIconfigurationEnums

< This [Camera](#) Link specific feature describes the configuration used by the camera. It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitization, Taps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera.

## Enumerator

**CIconfiguration\_Base** Standard base configuration described by the [Camera](#) Link standard.

**CIconfiguration\_Medium** Standard medium configuration described by the [Camera](#) Link standard.

**CIconfiguration\_Full** Standard full configuration described by the [Camera](#) Link standard.

**CIconfiguration\_DualBase** The camera streams the data from multiple taps (that do not fit in the standard base configuration) through two [Camera](#) Link base ports. It is responsibility of the application or frame grabber to reconstruct the full image. Only one of the ports (fixed) serves as the "master" for serial communication and triggering.

**CIconfiguration\_EightyBit** Standard 80-bit configuration with 10 taps of 8 bits or 8 taps of 10 bits, as described by the [Camera](#) Link standard.

**NUM\_CLCONFIGURATION**

## 8.8.2.40 enum CTimeSlotsCountEnums

< This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

## Enumerator

**CTimeSlotsCount\_One** One

**CTimeSlotsCount\_Two** Two

**CTimeSlotsCount\_Three** Three

**NUM\_CLTIMESLOTSCOUNT**

## 8.8.2.41 enum ColorTransformationSelectorEnums

< Selects which Color Transformation module is controlled by the various Color Transformation features

## Enumerator

**ColorTransformationSelector\_RGBtoRGB**

**ColorTransformationSelector\_RGBtoYUV**

**NUM\_COLORTRANSFORMATIONSELECTOR**

## 8.8.2.42 enum ColorTransformationValueSelectorEnums

< Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module

Enumerator

```

ColorTransformationValueSelector_Gain00
ColorTransformationValueSelector_Gain01
ColorTransformationValueSelector_Gain02
ColorTransformationValueSelector_Gain10
ColorTransformationValueSelector_Gain11
ColorTransformationValueSelector_Gain12
ColorTransformationValueSelector_Gain20
ColorTransformationValueSelector_Gain21
ColorTransformationValueSelector_Gain22
ColorTransformationValueSelector_Offset0
ColorTransformationValueSelector_Offset1
ColorTransformationValueSelector_Offset2
NUM_COLORTRANSFORMATIONVALUESELECTOR

```

## 8.8.2.43 enum CounterEventActivationEnums

< Selects the activation mode of the event to increment the Counter.

Enumerator

```

CounterEventActivation_LevelLow
CounterEventActivation_LevelHigh
CounterEventActivation_FallingEdge
CounterEventActivation_RisingEdge
CounterEventActivation_AnyEdge
NUM_COUNTEREVENTACTIVATION

```

## 8.8.2.44 enum CounterEventSourceEnums

< Selects the event that will increment the counter

Enumerator

```

CounterEventSource_Off Off
CounterEventSource_MHzTick MHzTick
CounterEventSource_Line0 Line0
CounterEventSource_Line1 Line1
CounterEventSource_Line2 Line2
CounterEventSource_Line3 Line3
CounterEventSource_UserOutput0 UserOutput0

```

**CounterEventSource\_UserOutput1** UserOutput1  
**CounterEventSource\_UserOutput2** UserOutput2  
**CounterEventSource\_UserOutput3** UserOutput3  
**CounterEventSource\_Counter0Start** Counter0Start  
**CounterEventSource\_Counter1Start** Counter1Start  
**CounterEventSource\_Counter0End** Counter0End  
**CounterEventSource\_Counter1End** Counter1End  
**CounterEventSource\_LogicBlock0** LogicBlock0  
**CounterEventSource\_LogicBlock1** LogicBlock1  
**CounterEventSource\_ExposureStart** ExposureStart  
**CounterEventSource\_ExposureEnd** ExposureEnd  
**CounterEventSource\_FrameTriggerWait** FrameTriggerWait  
**NUM\_COUNTEREVENTSOURCE**

#### 8.8.2.45 enum CounterResetActivationEnums

< Selects the Activation mode of the Counter Reset Source signal.

Enumerator

**CounterResetActivation\_LevelLow**  
**CounterResetActivation\_LevelHigh**  
**CounterResetActivation\_FallingEdge**  
**CounterResetActivation\_RisingEdge**  
**CounterResetActivation\_AnyEdge**  
**NUM\_COUNTERRESETACTIVATION**

#### 8.8.2.46 enum CounterResetSourceEnums

< Selects the signal that will be the source to reset the Counter.

Enumerator

**CounterResetSource\_Off** Off  
**CounterResetSource\_Line0** Line0  
**CounterResetSource\_Line1** Line1  
**CounterResetSource\_Line2** Line2  
**CounterResetSource\_Line3** Line3  
**CounterResetSource\_UserOutput0** UserOutput0  
**CounterResetSource\_UserOutput1** UserOutput1  
**CounterResetSource\_UserOutput2** UserOutput2  
**CounterResetSource\_UserOutput3** UserOutput3  
**CounterResetSource\_Counter0Start** Counter0Start  
**CounterResetSource\_Counter1Start** Counter1Start  
**CounterResetSource\_Counter0End** Counter0End  
**CounterResetSource\_Counter1End** Counter1End  
**CounterResetSource\_LogicBlock0** LogicBlock0  
**CounterResetSource\_LogicBlock1** LogicBlock1  
**CounterResetSource\_ExposureStart** ExposureStart  
**CounterResetSource\_ExposureEnd** ExposureEnd  
**CounterResetSource\_FrameTriggerWait** FrameTriggerWait  
**NUM\_COUNTERRESETSOURCE**

## 8.8.2.47 enum CounterSelectorEnums

< Selects which counter to configure

Enumerator

***CounterSelector\_Counter0***

***CounterSelector\_Counter1***

***NUM\_COUNTERSELECTOR***

## 8.8.2.48 enum CounterStatusEnums

< Returns the current status of the Counter.

Enumerator

***CounterStatus\_CounterIdle*** The counter is idle.

***CounterStatus\_CounterTriggerWait*** The counter is waiting for a start trigger.

***CounterStatus\_CounterActive*** The counter is counting for the specified duration.

***CounterStatus\_CounterCompleted*** The counter reached the CounterDuration count.

***CounterStatus\_CounterOverflow*** The counter reached its maximum possible count.

***NUM\_COUNTERSTATUS***

## 8.8.2.49 enum CounterTriggerActivationEnums

< Selects the activation mode of the trigger to start the Counter.

Enumerator

***CounterTriggerActivation\_LevelLow***

***CounterTriggerActivation\_LevelHigh***

***CounterTriggerActivation\_FallingEdge***

***CounterTriggerActivation\_RisingEdge***

***CounterTriggerActivation\_AnyEdge***

***NUM\_COUNTERTRIGGERACTIVATION***

## 8.8.2.50 enum CounterTriggerSourceEnums

< Selects the source of the trigger to start the counter

## Enumerator

**CounterTriggerSource\_Off** Off  
**CounterTriggerSource\_Line0** Line0  
**CounterTriggerSource\_Line1** Line1  
**CounterTriggerSource\_Line2** Line2  
**CounterTriggerSource\_Line3** Line3  
**CounterTriggerSource\_UserOutput0** UserOutput0  
**CounterTriggerSource\_UserOutput1** UserOutput1  
**CounterTriggerSource\_UserOutput2** UserOutput2  
**CounterTriggerSource\_UserOutput3** UserOutput3  
**CounterTriggerSource\_Counter0Start** Counter0Start  
**CounterTriggerSource\_Counter1Start** Counter1Start  
**CounterTriggerSource\_Counter0End** Counter0End  
**CounterTriggerSource\_Counter1End** Counter1End  
**CounterTriggerSource\_LogicBlock0** LogicBlock0  
**CounterTriggerSource\_LogicBlock1** LogicBlock1  
**CounterTriggerSource\_ExposureStart** ExposureStart  
**CounterTriggerSource\_ExposureEnd** ExposureEnd  
**CounterTriggerSource\_FrameTriggerWait** FrameTriggerWait  
**NUM\_COUNTERTRIGGERSOURCE**

## 8.8.2.51 enum CxpConnectionTestModeEnums

< Enables the test mode for an individual physical connection of the Device.

## Enumerator

**CxpConnectionTestMode\_Off** Off  
**CxpConnectionTestMode\_Mode1** Mode 1  
**NUM\_CXP\_CONNECTIONTESTMODE**

## 8.8.2.52 enum CxpLinkConfigurationEnums

< This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device. In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus.

## Enumerator

**CxpLinkConfiguration\_Auto** Sets Automatic discovery for the Link Configuration.  
**CxpLinkConfiguration\_CXP1\_X1** Force the Link to 1 Connection operating at CXP-1 speed (1.25 Gbps).



***CxpLinkConfiguration\_CXP2\_X1*** Force the Link to 1 Connection operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfiguration\_CXP3\_X1*** Force the Link to 1 Connection operating at CXP-3 speed (3.125 Gbps).  
  
***CxpLinkConfiguration\_CXP5\_X1*** Force the Link to 1 Connection operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfiguration\_CXP6\_X1*** Force the Link to 1 Connection operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfiguration\_CXP1\_X2*** Force the Link to 2 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfiguration\_CXP2\_X2*** Force the Link to 2 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfiguration\_CXP3\_X2*** Force the Link to 2 Connections operating at CXP-3 speed (3.125 Gbps).  
  
***CxpLinkConfiguration\_CXP5\_X2*** Force the Link to 2 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfiguration\_CXP6\_X2*** Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfiguration\_CXP1\_X3*** Force the Link to 3 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfiguration\_CXP2\_X3*** Force the Link to 3 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfiguration\_CXP3\_X3*** Force the Link to 3 Connections operating at CXP-3 speed (3.125 Gbps).  
  
***CxpLinkConfiguration\_CXP5\_X3*** Force the Link to 3 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfiguration\_CXP6\_X3*** Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfiguration\_CXP1\_X4*** Force the Link to 4 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfiguration\_CXP2\_X4*** Force the Link to 4 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfiguration\_CXP3\_X4*** Force the Link to 4 Connections operating at CXP-3 speed (3.125 Gbps).  
  
***CxpLinkConfiguration\_CXP5\_X4*** Force the Link to 4 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfiguration\_CXP6\_X4*** Force the Link to 4 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfiguration\_CXP1\_X5*** Force the Link to 5 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfiguration\_CXP2\_X5*** Force the Link to 5 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfiguration\_CXP3\_X5*** Force the Link to 5 Connections operating at CXP-3 speed (3.125 Gbps).  
  
***CxpLinkConfiguration\_CXP5\_X5*** Force the Link to 5 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfiguration\_CXP6\_X5*** Force the Link to 5 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfiguration\_CXP1\_X6*** Force the Link to 6 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfiguration\_CXP2\_X6*** Force the Link to 6 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfiguration\_CXP3\_X6*** Force the Link to 6 Connections operating at CXP-3 speed (3.125 Gbps).  
  
***CxpLinkConfiguration\_CXP5\_X6*** Force the Link to 6 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfiguration\_CXP6\_X6*** Force the Link to 6 Connections operating at CXP-6 speed (6.25 Gbps).  
***NUM\_CXPLINKCONFIGURATION***

#### 8.8.2.53 enum CxpLinkConfigurationPreferredEnums

< Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

##### Enumerator

***CxpLinkConfigurationPreferred\_CXP1\_X1*** 1 Connection operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP2\_X1*** 1 Connection operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationPreferred\_CXP3\_X1*** 1 Connection operating at CXP-3 speed (3.125 Gbps).

***CxpLinkConfigurationPreferred\_CXP5\_X1*** 1 Connection operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationPreferred\_CXP6\_X1*** 1 Connection operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP1\_X2*** 2 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP2\_X2*** 2 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationPreferred\_CXP3\_X2*** 2 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationPreferred\_CXP5\_X2*** 2 Connections operating at CXP-4 speed (5.00 Gbps).  
***CxpLinkConfigurationPreferred\_CXP6\_X2*** 3 Connections operating at CXP-5 speed (6.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP1\_X3*** 3 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP2\_X3*** 3 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationPreferred\_CXP3\_X3*** 3 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationPreferred\_CXP5\_X3*** 3 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationPreferred\_CXP6\_X3*** 3 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP1\_X4*** 4 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP2\_X4*** 4 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationPreferred\_CXP3\_X4*** 4 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationPreferred\_CXP5\_X4*** 4 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationPreferred\_CXP6\_X4*** 4 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP1\_X5*** 5 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP2\_X5*** 5 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationPreferred\_CXP3\_X5*** 5 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationPreferred\_CXP5\_X5*** 5 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationPreferred\_CXP6\_X5*** 5 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP1\_X6*** 6 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationPreferred\_CXP2\_X6*** 6 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationPreferred\_CXP3\_X6*** 6 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationPreferred\_CXP5\_X6*** 6 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationPreferred\_CXP6\_X6*** 6 Connections operating at CXP-6 speed (6.25 Gbps).  
**NUM\_CXPLINKCONFIGURATIONPREFERRED**

#### 8.8.2.54 enum CxpLinkConfigurationStatusEnums

< This feature indicates the current and active Link configuration used by the Device.

##### Enumerator

***CxpLinkConfigurationStatus\_None*** The Link configuration of the Device is unknown. Either the configuration operation has failed or there is nothing connected.  
***CxpLinkConfigurationStatus\_Pending*** The Device is in the process of configuring the Link. The Link cannot be used yet.  
***CxpLinkConfigurationStatus\_CXP1\_X1*** 1 Connection operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP2\_X1*** 1 Connection operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationStatus\_CXP3\_X1*** 1 Connection operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationStatus\_CXP5\_X1*** 1 Connection operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationStatus\_CXP6\_X1*** 1 Connection operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP1\_X2*** 2 Connections operating at CXP-1 speed (1.25 Gbps).

***CxpLinkConfigurationStatus\_CXP2\_X2*** 2 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationStatus\_CXP3\_X2*** 2 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationStatus\_CXP5\_X2*** 2 Connections operating at CXP-4 speed (5.00 Gbps).  
***CxpLinkConfigurationStatus\_CXP6\_X2*** 3 Connections operating at CXP-5 speed (6.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP1\_X3*** 3 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP2\_X3*** 3 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationStatus\_CXP3\_X3*** 3 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationStatus\_CXP5\_X3*** 3 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationStatus\_CXP6\_X3*** 3 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP1\_X4*** 4 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP2\_X4*** 4 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationStatus\_CXP3\_X4*** 4 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationStatus\_CXP5\_X4*** 4 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationStatus\_CXP6\_X4*** 4 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP1\_X5*** 5 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP2\_X5*** 5 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationStatus\_CXP3\_X5*** 5 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationStatus\_CXP5\_X5*** 5 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationStatus\_CXP6\_X5*** 5 Connections operating at CXP-6 speed (6.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP1\_X6*** 6 Connections operating at CXP-1 speed (1.25 Gbps).  
***CxpLinkConfigurationStatus\_CXP2\_X6*** 6 Connections operating at CXP-2 speed (2.50 Gbps).  
***CxpLinkConfigurationStatus\_CXP3\_X6*** 6 Connections operating at CXP-3 speed (3.125 Gbps).  
***CxpLinkConfigurationStatus\_CXP5\_X6*** 6 Connections operating at CXP-5 speed (5.00 Gbps).  
***CxpLinkConfigurationStatus\_CXP6\_X6*** 6 Connections operating at CXP-6 speed (6.25 Gbps).  
***NUM\_CXPLINKCONFIGURATIONSTATUS***

#### 8.8.2.55 enum CxpPoCxpStatusEnums

< Returns the Power over CoaXPress (PoCXP) status of the Device.

Enumerator

***CxpPoCxpStatus\_Auto*** Normal automatic PoCXP operation.  
***CxpPoCxpStatus\_Off*** PoCXP is forced off.  
***CxpPoCxpStatus\_Tripped*** The Link has shut down because of an over-current trip.  
***NUM\_CXPPOCXPSTATUS***

#### 8.8.2.56 enum DecimationHorizontalModeEnums

< The mode used to reduce the horizontal resolution when DecimationHorizontal is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Enumerator

***DecimationHorizontalMode\_Discard*** The value of every Nth pixel is kept, others are discarded.  
***NUM\_DECIMATIONHORIZONTALMODE***

## 8.8.2.57 enum DecimationSelectorEnums

< Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.

## Enumerator

**DecimationSelector\_All** The total amount of decimation to be performed on the captured image data.

**DecimationSelector\_Sensor** The portion of decimation to be performed on the sensor directly. Currently this is the only decimation layer available and hence is identical to the "All" layer. All decimation modification should therefore be done via the "All" layer only.

**NUM\_DECIMATIONSELECTOR**

## 8.8.2.58 enum DecimationVerticalModeEnums

< The mode used to reduce the vertical resolution when DecimationVertical is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

## Enumerator

**DecimationVerticalMode\_Discard** The value of every Nth pixel is kept, others are discarded.

**NUM\_DECIMATIONVERTICALMODE**

## 8.8.2.59 enum DefectCorrectionModeEnums

< Controls the method used for replacing defective pixels.

## Enumerator

**DefectCorrectionMode\_Average** Pixels are replaced with the average of their neighbours. This is the normal mode of operation.

**DefectCorrectionMode\_Highlight** Pixels are replaced with the maximum pixel value (i.e., 255 for 8-bit images). Can be used for debugging the table.

**DefectCorrectionMode\_Zero** Pixels are replaced by the value zero. Can be used for testing the table.

**NUM\_DEFECTCORRECTIONMODE**

## 8.8.2.60 enum DeinterlacingEnums

< Controls how the device performs de-interlacing.

## Enumerator

**Deinterlacing\_Off** The device doesn't perform de-interlacing.

**Deinterlacing\_LineDuplication** The device performs de-interlacing by outputting each line of each field twice.

**Deinterlacing\_Weave** The device performs de-interlacing by interleaving the lines of all fields.

**NUM\_DEINTERLACING**

## 8.8.2.61 enum DeviceCharacterSetEnums

< Character set used by the strings of the device's bootstrap registers.

Enumerator

***DeviceCharacterSet\_UTF8***  
***DeviceCharacterSet\_ASCII***  
***NUM\_DEVICECHARACTERSET***

## 8.8.2.62 enum DeviceClockSelectorEnums

< Selects the clock frequency to access from the device.

Enumerator

***DeviceClockSelector\_Sensor*** Clock frequency of the image sensor of the camera.  
***DeviceClockSelector\_SensorDigitization*** Clock frequency of the camera A/D conversion stage.  
***DeviceClockSelector\_CameraLink*** Frequency of the [Camera](#) Link clock.  
***NUM\_DEVICECLOCKSELECTOR***

## 8.8.2.63 enum DeviceConnectionStatusEnums

< Indicates the status of the specified Connection.

Enumerator

***DeviceConnectionStatus\_Active*** Connection is in use.  
***DeviceConnectionStatus\_Inactive*** Connection is not in use.  
***NUM\_DEVICECONNECTIONSTATUS***

## 8.8.2.64 enum DeviceIndicatorModeEnums

< Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).

Enumerator

***DeviceIndicatorMode\_Inactive***  
***DeviceIndicatorMode\_Active***  
***DeviceIndicatorMode\_ErrorStatus***  
***NUM\_DEVICEINDICATORMODE***

#### 8.8.2.65 enum DeviceLinkHeartbeatModeEnums

< Activate or deactivate the Link's heartbeat.

Enumerator

***DeviceLinkHeartbeatMode\_On*** Enables the Link heartbeat.

***DeviceLinkHeartbeatMode\_Off*** Disables the Link heartbeat.

***NUM\_DEVICELINKHEARTBEATMODE***

#### 8.8.2.66 enum DeviceLinkThroughputLimitModeEnums

< Controls if the DeviceLinkThroughputLimit is active. When disabled, lower level TL specific features are expected to control the throughput. When enabled, DeviceLinkThroughputLimit controls the overall throughput.

Enumerator

***DeviceLinkThroughputLimitMode\_On*** Enables the DeviceLinkThroughputLimit feature.

***DeviceLinkThroughputLimitMode\_Off*** Disables the DeviceLinkThroughputLimit feature.

***NUM\_DEVICELINKTHROUGHPUTLIMITMODE***

#### 8.8.2.67 enum DevicePowerSupplySelectorEnums

< Selects the power supply source to control or read.

Enumerator

***DevicePowerSupplySelector\_External***

***NUM\_DEVICEPOWERSUPPLYSELECTOR***

#### 8.8.2.68 enum DeviceRegistersEndiannessEnums

< Endianness of the registers of the device.

Enumerator

***DeviceRegistersEndianness\_Little***

***DeviceRegistersEndianness\_Big***

***NUM\_DEVICEREGISTERSENDIANNES***

#### 8.8.2.69 enum DeviceScanTypeEnums

< Scan type of the sensor of the device.

Enumerator

***DeviceScanType\_Areascan***

***NUM\_DEVICESCANTYPE***

## 8.8.2.70 enum DeviceSerialPortBaudRateEnums

< This feature controls the baud rate used by the selected serial port.

## Enumerator

**DeviceSerialPortBaudRate\_Baud9600** Serial port speed of 9600 baud.  
**DeviceSerialPortBaudRate\_Baud19200** Serial port speed of 19200 baud.  
**DeviceSerialPortBaudRate\_Baud38400** Serial port speed of 38400 baud.  
**DeviceSerialPortBaudRate\_Baud57600** Serial port speed of 57600 baud.  
**DeviceSerialPortBaudRate\_Baud115200** Serial port speed of 115200 baud.  
**DeviceSerialPortBaudRate\_Baud230400** Serial port speed of 230400 baud.  
**DeviceSerialPortBaudRate\_Baud460800** Serial port speed of 460800 baud.  
**DeviceSerialPortBaudRate\_Baud921600** Serial port speed of 921600 baud.  
**NUM\_DEVICESERIALPORTBAUDRATE**

## 8.8.2.71 enum DeviceSerialPortSelectorEnums

< Selects which serial port of the device to control.

## Enumerator

**DeviceSerialPortSelector\_CameraLink** Serial port associated to the [Camera](#) link connection.  
**NUM\_DEVICESERIALPORTSELECTOR**

## 8.8.2.72 enum DeviceStreamChannelEndiannessEnums

< Endianness of multi-byte pixel data for this stream.

## Enumerator

**DeviceStreamChannelEndianness\_Big** Stream channel data is big Endian.  
**DeviceStreamChannelEndianness\_Little** Stream channel data is little Endian.  
**NUM\_DEVICESTREAMCHANNELENDIANNESS**

## 8.8.2.73 enum DeviceStreamChannelTypeEnums

< Reports the type of the stream channel.

## Enumerator

**DeviceStreamChannelType\_Transmitter** Data stream transmitter channel.  
**DeviceStreamChannelType\_Receiver** Data stream receiver channel.  
**NUM\_DEVICESTREAMCHANNELTYPE**

## 8.8.2.74 enum DeviceTapGeometryEnums

< This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

## Enumerator

**DeviceTapGeometry\_Geometry\_1X\_1Y** Geometry\_1X\_1Y  
**DeviceTapGeometry\_Geometry\_1X2\_1Y** Geometry\_1X2\_1Y  
**DeviceTapGeometry\_Geometry\_1X2\_1Y2** Geometry\_1X2\_1Y2  
**DeviceTapGeometry\_Geometry\_2X\_1Y** Geometry\_2X\_1Y  
**DeviceTapGeometry\_Geometry\_2X\_1Y2Geometry\_2XE\_1Y** Geometry\_2X\_1Y2Geometry\_2XE\_1Y  
**DeviceTapGeometry\_Geometry\_2XE\_1Y2** Geometry\_2XE\_1Y2  
**DeviceTapGeometry\_Geometry\_2XM\_1Y** Geometry\_2XM\_1Y  
**DeviceTapGeometry\_Geometry\_2XM\_1Y2** Geometry\_2XM\_1Y2  
**DeviceTapGeometry\_Geometry\_1X\_1Y2** Geometry\_1X\_1Y2  
**DeviceTapGeometry\_Geometry\_1X\_2YE** Geometry\_1X\_2YE  
**DeviceTapGeometry\_Geometry\_1X3\_1Y** Geometry\_1X3\_1Y  
**DeviceTapGeometry\_Geometry\_3X\_1Y** Geometry\_3X\_1Y  
**DeviceTapGeometry\_Geometry\_1X** Geometry\_1X  
**DeviceTapGeometry\_Geometry\_1X2** Geometry\_1X2  
**DeviceTapGeometry\_Geometry\_2X** Geometry\_2X  
**DeviceTapGeometry\_Geometry\_2XE** Geometry\_2XE  
**DeviceTapGeometry\_Geometry\_2XM** Geometry\_2XM  
**DeviceTapGeometry\_Geometry\_1X3** Geometry\_1X3  
**DeviceTapGeometry\_Geometry\_3X** Geometry\_3X  
**DeviceTapGeometry\_Geometry\_1X4\_1Y** Geometry\_1X4\_1Y  
**DeviceTapGeometry\_Geometry\_4X\_1Y** Geometry\_4X\_1Y  
**DeviceTapGeometry\_Geometry\_2X2\_1Y** Geometry\_2X2\_1Y  
**DeviceTapGeometry\_Geometry\_2X2E\_1YGeometry\_2X2M\_1Y** Geometry\_2X2E\_1YGeometry\_2X2M↵  
 \_1Y  
**DeviceTapGeometry\_Geometry\_1X2\_2YE** Geometry\_1X2\_2YE  
**DeviceTapGeometry\_Geometry\_2X\_2YE** Geometry\_2X\_2YE  
**DeviceTapGeometry\_Geometry\_2XE\_2YE** Geometry\_2XE\_2YE  
**DeviceTapGeometry\_Geometry\_2XM\_2YE** Geometry\_2XM\_2YE  
**DeviceTapGeometry\_Geometry\_1X4** Geometry\_1X4  
**DeviceTapGeometry\_Geometry\_4X** Geometry\_4X  
**DeviceTapGeometry\_Geometry\_2X2** Geometry\_2X2  
**DeviceTapGeometry\_Geometry\_2X2E** Geometry\_2X2E  
**DeviceTapGeometry\_Geometry\_2X2M** Geometry\_2X2M  
**DeviceTapGeometry\_Geometry\_1X8\_1Y** Geometry\_1X8\_1Y  
**DeviceTapGeometry\_Geometry\_8X\_1Y** Geometry\_8X\_1Y  
**DeviceTapGeometry\_Geometry\_4X2\_1Y** Geometry\_4X2\_1Y  
**DeviceTapGeometry\_Geometry\_2X2E\_2YE** Geometry\_2X2E\_2YE  
**DeviceTapGeometry\_Geometry\_1X8** Geometry\_1X8  
**DeviceTapGeometry\_Geometry\_8X** Geometry\_8X



***DeviceTapGeometry\_Geometry\_4X2*** Geometry\_4X2  
***DeviceTapGeometry\_Geometry\_4X2E*** Geometry\_4X2E  
***DeviceTapGeometry\_Geometry\_4X2E\_1Y*** Geometry\_4X2E\_1Y  
***DeviceTapGeometry\_Geometry\_1X10\_1Y*** Geometry\_1X10\_1Y  
***DeviceTapGeometry\_Geometry\_10X\_1Y*** Geometry\_10X\_1Y  
***DeviceTapGeometry\_Geometry\_1X10*** Geometry\_1X10  
***DeviceTapGeometry\_Geometry\_10X*** Geometry\_10X  
***NUM\_DEVICETAPGEOMETRY***

#### 8.8.2.75 enum DeviceTemperatureSelectorEnums

< Selects the location within the device, where the temperature will be measured.

Enumerator

***DeviceTemperatureSelector\_Sensor***  
***NUM\_DEVICETEMPERATURESELECTOR***

#### 8.8.2.76 enum DeviceTLTypeEnums

< Transport Layer type of the device.

Enumerator

***DeviceTLType\_GigEVision***  
***DeviceTLType\_CameraLink***  
***DeviceTLType\_CameraLinkHS***  
***DeviceTLType\_CoaXPress***  
***DeviceTLType\_USB3Vision***  
***DeviceTLType\_Custom***  
***NUM\_DEVICETLTYPE***

#### 8.8.2.77 enum DeviceTypeEnums

< Returns the device type.

Enumerator

***DeviceType\_Transmitter*** Data stream transmitter device.  
***DeviceType\_Receiver*** Data stream receiver device.  
***DeviceType\_Transceiver*** Data stream receiver and transmitter device.  
***DeviceType\_Peripheral*** Controllable device (with no data stream handling).  
***NUM\_DEVICETYPE***

## 8.8.2.78 enum EncoderModeEnums

< Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.

Enumerator

**EncoderMode\_FourPhase** The counter increments or decrements 1 for every full quadrature cycle with jitter filtering.

**EncoderMode\_HighResolution** The counter increments or decrements every quadrature phase for high resolution counting, but without jitter filtering.

**NUM\_ENCODERMODE**

## 8.8.2.79 enum EncoderOutputModeEnums

< Selects the conditions for the Encoder interface to generate a valid Encoder output signal.

Enumerator

**EncoderOutputMode\_Off** No output pulse are generated.

**EncoderOutputMode\_PositionUp** Output pulses are generated at all new positions in the positive direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started.

**EncoderOutputMode\_PositionDown** Output pulses are generated at all new positions in the negative direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started.

**EncoderOutputMode\_DirectionUp** Output pulses are generated at all position increments in the positive direction while ignoring negative direction motion.

**EncoderOutputMode\_DirectionDown** Output pulses are generated at all position increments in the negative direction while ignoring positive direction motion.

**EncoderOutputMode\_Motion** Output pulses are generated at all motion increments in both directions.

**NUM\_ENCODEROUTPUTMODE**

## 8.8.2.80 enum EncoderResetActivationEnums

< Selects the Activation mode of the Encoder Reset Source signal.

Enumerator

**EncoderResetActivation\_RisingEdge** Resets the Encoder on the Rising Edge of the signal.

**EncoderResetActivation\_FallingEdge** Resets the Encoder on the Falling Edge of the signal.

**EncoderResetActivation\_AnyEdge** Resets the Encoder on the Falling or rising Edge of the selected signal.

**EncoderResetActivation\_LevelHigh** Resets the Encoder as long as the selected signal level is High.

**EncoderResetActivation\_LevelLow** Resets the Encoder as long as the selected signal level is Low.

**NUM\_ENCODERRESETACTIVATION**

## 8.8.2.81 enum EncoderResetSourceEnums

< Selects the signals that will be the source to reset the Encoder.

Enumerator

**EncoderResetSource\_Off** Disable the Encoder Reset trigger.

**EncoderResetSource\_AcquisitionTrigger** Resets with the reception of the Acquisition Trigger.

**EncoderResetSource\_AcquisitionStart** Resets with the reception of the Acquisition Start.

**EncoderResetSource\_AcquisitionEnd** Resets with the reception of the Acquisition End.

**EncoderResetSource\_FrameTrigger** Resets with the reception of the Frame Start Trigger.

**EncoderResetSource\_FrameStart** Resets with the reception of the Frame Start.

**EncoderResetSource\_FrameEnd** Resets with the reception of the Frame End.

**EncoderResetSource\_ExposureStart** Resets with the reception of the Exposure Start.

**EncoderResetSource\_ExposureEnd** Resets with the reception of the Exposure End.

**EncoderResetSource\_Line0** Resets by the chosen I/O Line.

**EncoderResetSource\_Line1** Resets by the chosen I/O Line.

**EncoderResetSource\_Line2** Resets by the chosen I/O Line.

**EncoderResetSource\_Counter0Start** Resets with the reception of the Counter Start.

**EncoderResetSource\_Counter1Start** Resets with the reception of the Counter Start.

**EncoderResetSource\_Counter2Start** Resets with the reception of the Counter Start.

**EncoderResetSource\_Counter0End** Resets with the reception of the Counter End.

**EncoderResetSource\_Counter1End** Resets with the reception of the Counter End.

**EncoderResetSource\_Counter2End** Resets with the reception of the Counter End.

**EncoderResetSource\_Timer0Start** Resets with the reception of the Timer Start.

**EncoderResetSource\_Timer1Start** Resets with the reception of the Timer Start.

**EncoderResetSource\_Timer2Start** Resets with the reception of the Timer Start.

**EncoderResetSource\_Timer0End** Resets with the reception of the Timer End.

**EncoderResetSource\_Timer1End** Resets with the reception of the Timer End.

**EncoderResetSource\_Timer2End** Resets with the reception of the Timer End.

**EncoderResetSource\_UserOutput0** Resets by the chosen User Output bit.

**EncoderResetSource\_UserOutput1** Resets by the chosen User Output bit.

**EncoderResetSource\_UserOutput2** Resets by the chosen User Output bit.

**EncoderResetSource\_SoftwareSignal0** Resets on the reception of the Software Signal.

**EncoderResetSource\_SoftwareSignal1** Resets on the reception of the Software Signal.

**EncoderResetSource\_SoftwareSignal2** Resets on the reception of the Software Signal.

**EncoderResetSource\_Action0** Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).

**EncoderResetSource\_Action1** Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).

**EncoderResetSource\_Action2** Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).

**EncoderResetSource\_LinkTrigger0** Resets on the reception of the chosen Link Trigger (received from the transport layer).

**EncoderResetSource\_LinkTrigger1** Resets on the reception of the chosen Link Trigger (received from the transport layer).

**EncoderResetSource\_LinkTrigger2** Resets on the reception of the chosen Link Trigger (received from the transport layer).

**NUM\_ENCODERRESETSOURCE**

## 8.8.2.82 enum EncoderSelectorEnums

< Selects which Encoder to configure.

Enumerator

**EncoderSelector\_Encoder0** Selects Encoder 0.

**EncoderSelector\_Encoder1** Selects Encoder 1.

**EncoderSelector\_Encoder2** Selects Encoder 2.

**NUM\_ENCODERSELECTOR**

## 8.8.2.83 enum EncoderSourceAEnums

< Selects the signal which will be the source of the A input of the Encoder.

Enumerator

**EncoderSourceA\_Off** Counter is stopped.

**EncoderSourceA\_Line0** Encoder Forward input is taken from the chosen I/O Line.

**EncoderSourceA\_Line1** Encoder Forward input is taken from the chosen I/O Line.

**EncoderSourceA\_Line2** Encoder Forward input is taken from the chosen I/O Line.

**NUM\_ENCODERSOURCEA**

## 8.8.2.84 enum EncoderSourceBEnums

< Selects the signal which will be the source of the B input of the Encoder.

Enumerator

**EncoderSourceB\_Off** Counter is stopped.

**EncoderSourceB\_Line0** Encoder Reverse input is taken from the chosen I/O Line..

**EncoderSourceB\_Line1** Encoder Reverse input is taken from the chosen I/O Line..

**EncoderSourceB\_Line2** Encoder Reverse input is taken from the chosen I/O Line..

**NUM\_ENCODERSOURCEB**

## 8.8.2.85 enum EncoderStatusEnums

< Returns the motion status of the encoder.

Enumerator

**EncoderStatus\_EncoderUp** The encoder counter last incremented.

**EncoderStatus\_EncoderDown** The encoder counter last decremented.

**EncoderStatus\_EncoderIdle** The encoder is not active.

**EncoderStatus\_EncoderStatic** No motion within the EncoderTimeout time.

**NUM\_ENCODERSTATUS**

## 8.8.2.86 enum EventNotificationEnums

< Enables/Disables the selected event.

Enumerator

***EventNotification\_On***  
***EventNotification\_Off***  
***NUM\_EVENTNOTIFICATION***

## 8.8.2.87 enum EventSelectorEnums

< Selects which [Event](#) to enable or disable.

Enumerator

***EventSelector\_Error***  
***EventSelector\_ExposureEnd***  
***EventSelector\_SerialPortReceive***  
***NUM\_EVENTSELECTOR***

## 8.8.2.88 enum ExposureActiveModeEnums

< Control sensor active exposure mode.

Enumerator

***ExposureActiveMode\_Line1***  
***ExposureActiveMode\_AnyPixels***  
***ExposureActiveMode\_AllPixels***  
***NUM\_EXPOSUREACTIVEMODE***

## 8.8.2.89 enum ExposureAutoEnums

< Sets the automatic exposure mode

Enumerator

***ExposureAuto\_Off*** Exposure time is manually controlled using ExposureTime  
***ExposureAuto\_Once*** Exposure time is adapted once by the device. Once it has converged, it returns to the Off state.  
***ExposureAuto\_Continuous*** Exposure time is constantly adapted by the device to maximize the dynamic range.  
***NUM\_EXPOSUREAUTO***

## 8.8.2.90 enum ExposureModeEnums

< Sets the operation mode of the Exposure.

## Enumerator

**ExposureMode\_Timed** Timed exposure. The exposure time is set using the ExposureTime or ExposureTimeAuto features and the exposure starts with the FrameStart or LineStart.

**ExposureMode\_TriggerWidth** Uses the width of the current Frame trigger signal pulse to control the exposure time.

**NUM\_EXPOSUREMODE**

## 8.8.2.91 enum ExposureTimeModeEnums

< Sets the configuration mode of the ExposureTime feature.

## Enumerator

**ExposureTimeMode\_Common** The exposure time is common to all the color components. The common ExposureTime value to use can be set selecting it with ExposureTimeSelector[Common].

**ExposureTimeMode\_Individual** The exposure time is individual for each color component. Each individual ExposureTime values to use can be set by selecting them with ExposureTimeSelector.

**NUM\_EXPOSURETIMEMODE**

## 8.8.2.92 enum ExposureTimeSelectorEnums

< Selects which exposure time is controlled by the ExposureTime feature. This allows for independent control over the exposure components.

## Enumerator

**ExposureTimeSelector\_Common** Selects the common ExposureTime.

**ExposureTimeSelector\_Red** Selects the red common ExposureTime.

**ExposureTimeSelector\_Green** Selects the green ExposureTime.

**ExposureTimeSelector\_Blue** Selects the blue ExposureTime.

**ExposureTimeSelector\_Cyan** Selects the cyan common ExposureTime.

**ExposureTimeSelector\_Magenta** Selects the magenta ExposureTime.

**ExposureTimeSelector\_Yellow** Selects the yellow ExposureTime.

**ExposureTimeSelector\_Infrared** Selects the infrared ExposureTime.

**ExposureTimeSelector\_Ultraviolet** Selects the ultraviolet ExposureTime.

**ExposureTimeSelector\_Stage1** Selects the first stage ExposureTime.

**ExposureTimeSelector\_Stage2** Selects the second stage ExposureTime.

**NUM\_EXPOSURETIMESELECTOR**

## 8.8.2.93 enum FileOpenModeEnums

< The mode of the file when it is opened. The file can be opened for reading, writing or both. This must be set before opening the file.

Enumerator

***FileOpenMode\_Read***  
***FileOpenMode\_Write***  
***FileOpenMode\_ReadWrite***  
***NUM\_FILEOPENMODE***

## 8.8.2.94 enum FileOperationSelectorEnums

< Sets operation to execute on the selected file when the execute command is given.

Enumerator

***FileOperationSelector\_Open***  
***FileOperationSelector\_Close***  
***FileOperationSelector\_Read***  
***FileOperationSelector\_Write***  
***FileOperationSelector\_Delete***  
***NUM\_FILEOPERATIONSELECTOR***

## 8.8.2.95 enum FileOperationStatusEnums

< Represents the file operation execution status.

Enumerator

***FileOperationStatus\_Success*** File Operation was sucessful.  
***FileOperationStatus\_Failure*** File Operation failed.  
***FileOperationStatus\_Overflow*** An overflow occurred while executing the File Operation.  
***NUM\_FILEOPERATIONSTATUS***

## 8.8.2.96 enum FileSelectorEnums

< Selects which file is being operated on. This must be set before performing any file operations.

Enumerator

***FileSelector\_UserSetDefault***  
***FileSelector\_UserSet0***  
***FileSelector\_UserSet1***  
***FileSelector\_UserFile1***  
***FileSelector\_SerialPort0***  
***NUM\_FILESELECTOR***

### 8.8.2.97 enum GainAutoBalanceEnums

< Sets the mode for automatic gain balancing between the sensor color channels or taps. The gain coefficients of each channel or tap are adjusted so they are matched.

#### Enumerator

**GainAutoBalance\_Off** Gain tap balancing is user controlled using Gain .

**GainAutoBalance\_Once** Gain tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.

**GainAutoBalance\_Continuous** Gain tap balancing is constantly adjusted by the device.

**NUM\_GAINAUTOBALANCE**

### 8.8.2.98 enum GainAutoEnums

< Sets the automatic gain mode. Set to Off for manual control. Set to Once for a single automatic adjustment then return to Off. Set to Continuous for constant adjustment. In automatic modes, the camera adjusts the gain to maximize the dynamic range.

#### Enumerator

**GainAuto\_Off** Gain is manually controlled

**GainAuto\_Once** Gain is adapted once by the device. Once it has converged, it returns to the Off state.

**GainAuto\_Continuous** Gain is constantly adapted by the device to maximize the dynamic range.

**NUM\_GAINAUTO**

### 8.8.2.99 enum GainSelectorEnums

< Selects which gain to control. The All selection is a total amplification across all channels (or taps).

#### Enumerator

**GainSelector\_All**

**NUM\_GAINSELECTOR**

### 8.8.2.100 enum GevCCPEnums

< Controls the device access privilege of an application.

#### Enumerator

**GevCCP\_OpenAccess**

**GevCCP\_ExclusiveAccess**

**GevCCP\_ControlAccess**

**NUM\_GEVCCP**



8.8.2.101 enum `GevCurrentPhysicalLinkConfigurationEnums`

< Indicates the current physical link configuration of the device.

## Enumerator

***GevCurrentPhysicalLinkConfiguration\_SingleLink*** Single Link  
***GevCurrentPhysicalLinkConfiguration\_MultiLink*** Multi Link  
***GevCurrentPhysicalLinkConfiguration\_StaticLAG*** Static LAG  
***GevCurrentPhysicalLinkConfiguration\_DynamicLAG*** Dynamic LAG  
***NUM\_GEVCURRENTPHYSICALLINKCONFIGURATION***

8.8.2.102 enum `GevGVCPExtendedStatusCodesSelectorEnums`

< Selects the GigE Vision version to control extended status codes for.

## Enumerator

***GevGVCPExtendedStatusCodesSelector\_Version1\_1*** Version 1 1  
***GevGVCPExtendedStatusCodesSelector\_Version2\_0*** Version 2 0  
***NUM\_GEVGVCPEXTENDEDSTATUSCODESSELECTOR***

8.8.2.103 enum `GevGVSPExtendedIDModeEnums`

< Enables the extended IDs mode.

## Enumerator

***GevGVSPExtendedIDMode\_Off*** Off  
***GevGVSPExtendedIDMode\_On*** On  
***NUM\_GEVGVSPEXTENDEDIDMODE***

8.8.2.104 enum `GevIEEE1588ClockAccuracyEnums`

< Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.

## Enumerator

***GevIEEE1588ClockAccuracy\_Unknown*** Unknown Accuracy  
***NUM\_GEVIIEEE1588CLOCKACCURACY***

## 8.8.2.105 enum GevIEEE1588ModeEnums

< Provides the mode of the IEEE 1588 clock.

Enumerator

***GevIEEE1588Mode\_Auto*** Automatic  
***GevIEEE1588Mode\_SlaveOnly*** Slave Only  
***NUM\_GEVIEEE1588MODE***

## 8.8.2.106 enum GevIEEE1588StatusEnums

< Provides the status of the IEEE 1588 clock.

Enumerator

***GevIEEE1588Status\_Initializing*** Initializing  
***GevIEEE1588Status\_Faulty*** Faulty  
***GevIEEE1588Status\_Disabled*** Disabled  
***GevIEEE1588Status\_Listening*** Listening  
***GevIEEE1588Status\_PreMaster*** Pre Master  
***GevIEEE1588Status\_Master*** Master  
***GevIEEE1588Status\_Passive*** Passive  
***GevIEEE1588Status\_Uncalibrated*** Uncalibrated  
***GevIEEE1588Status\_Slave*** Slave  
***NUM\_GEVIEEE1588STATUS***

## 8.8.2.107 enum GevIPConfigurationStatusEnums

< Reports the current IP configuration status.

Enumerator

***GevIPConfigurationStatus\_None*** None  
***GevIPConfigurationStatus\_PersistentIP*** Persistent IP  
***GevIPConfigurationStatus\_DHCP*** DHCP  
***GevIPConfigurationStatus\_LLA*** LLA  
***GevIPConfigurationStatus\_ForceIP*** Force IP  
***NUM\_GEVIPCONFIGURATIONSTATUS***

## 8.8.2.108 enum GevPhysicalLinkConfigurationEnums

< Controls the principal physical link configuration to use on next restart/power-up of the device.

Enumerator

***GevPhysicalLinkConfiguration\_SingleLink*** Single Link  
***GevPhysicalLinkConfiguration\_MultiLink*** Multi Link  
***GevPhysicalLinkConfiguration\_StaticLAG*** Static LAG  
***GevPhysicalLinkConfiguration\_DynamicLAG*** Dynamic LAG  
***NUM\_GEVPHYSICALLINKCONFIGURATION***

## 8.8.2.109 enum GevSupportedOptionSelectorEnums

< Selects the GEV option to interrogate for existing support.

Enumerator

*GevSupportedOptionSelector\_UserDefinedName*  
*GevSupportedOptionSelector\_SerialNumber*  
*GevSupportedOptionSelector\_HeartbeatDisable*  
*GevSupportedOptionSelector\_LinkSpeed*  
*GevSupportedOptionSelector\_CCPApplicationSocket*  
*GevSupportedOptionSelector\_ManifestTable*  
*GevSupportedOptionSelector\_TestData*  
*GevSupportedOptionSelector\_DiscoveryAckDelay*  
*GevSupportedOptionSelector\_DiscoveryAckDelayWritable*  
*GevSupportedOptionSelector\_ExtendedStatusCodes*  
*GevSupportedOptionSelector\_Action*  
*GevSupportedOptionSelector\_PendingAck*  
*GevSupportedOptionSelector\_EventData*  
*GevSupportedOptionSelector\_Event*  
*GevSupportedOptionSelector\_PacketResend*  
*GevSupportedOptionSelector\_WriteMem*  
*GevSupportedOptionSelector\_CommandsConcatenation*  
*GevSupportedOptionSelector\_IPConfigurationLLA*  
*GevSupportedOptionSelector\_IPConfigurationDHCP*  
*GevSupportedOptionSelector\_IPConfigurationPersistentIP*  
*GevSupportedOptionSelector\_StreamChannelSourceSocket*  
*GevSupportedOptionSelector\_MessageChannelSourceSocket*  
*NUM\_GEVSUPPORTEDOPTIONSELECTOR*

## 8.8.2.110 enum ImageComponentSelectorEnums

< Selects a component to activate data streaming from.

Enumerator

*ImageComponentSelector\_Intensity* The acquisition of intensity of the reflected light is controlled.  
*ImageComponentSelector\_Color* The acquisition of color of the reflected light is controlled  
*ImageComponentSelector\_Infrared* The acquisition of non-visible infrared light is controlled.  
*ImageComponentSelector\_Ultraviolet* The acquisition of non-visible ultraviolet light is controlled.  
*ImageComponentSelector\_Range* The acquisition of range (distance) data is controlled. The data produced may be only range (2.5D) or a point cloud 3D coordinates depending on the Scan3dControl.  
*ImageComponentSelector\_Disparity* The acquisition of stereo camera disparity data is controlled. Disparity is a more specific range format approximately inversely proportional to distance. Disparity is typically given in pixel units.

***ImageComponentSelector\_Confidence*** The acquisition of confidence map of the acquired image is controlled. Confidence data may be binary (0 - invalid) or an integer where 0 is invalid and increasing value is increased confidence in the data in the corresponding pixel. If floating point representation is used the confidence image is normalized to the range [0,1], for integer representation the maximum possible integer represents maximum confidence.

***ImageComponentSelector\_Scatter*** The acquisition of data measuring how much light is scattered around the reflected light. In processing this is used as an additional intensity image, often together with the standard intensity.

***NUM\_IMAGECOMPONENTSELECTOR***

#### 8.8.2.111 enum ImageCompressionJPEGFormatOptionEnums

< When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

Enumerator

***ImageCompressionJPEGFormatOption\_Lossless*** Selects lossless JPEG compression based on a predictive coding model.

***ImageCompressionJPEGFormatOption\_BaselineStandard*** Indicates this is a baseline sequential (single-scan) DCT-based JPEG.

***ImageCompressionJPEGFormatOption\_BaselineOptimized*** Provides optimized color and slightly better compression than baseline standard by using custom Huffman tables optimized after statistical analysis of the image content.

***ImageCompressionJPEGFormatOption\_Progressive*** Indicates this is a progressive (multi-scan) DCT-based JPEG.

***NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION***

#### 8.8.2.112 enum ImageCompressionModeEnums

<

Enumerator

***ImageCompressionMode\_Off***

***ImageCompressionMode\_Lossless***

***NUM\_IMAGECOMPRESSIONMODE***

#### 8.8.2.113 enum ImageCompressionRateOptionEnums

< Two rate controlling options are offered: fixed bit rate or fixed quality. The exact implementation to achieve one or the other is vendor-specific.

Enumerator

***ImageCompressionRateOption\_FixBitrate*** Output stream follows a constant bit rate. Allows easy bandwidth management on the link.

***ImageCompressionRateOption\_FixQuality*** Output stream has a constant image quality. Can be used when image processing algorithms are sensitive to image degradation caused by excessive data compression.

***NUM\_IMAGECOMPRESSIONRATEOPTION***

## 8.8.2.114 enum LineFormatEnums

< Displays the current electrical format of the selected physical input or output Line.

Enumerator

***LineFormat\_NoConnect***  
***LineFormat\_TriState***  
***LineFormat\_TTL***  
***LineFormat\_LVDS***  
***LineFormat\_RS422***  
***LineFormat\_OptoCoupled***  
***LineFormat\_OpenDrain***  
***NUM\_LINEFORMAT***

## 8.8.2.115 enum LineInputFilterSelectorEnums

< Selects the kind of input filter to configure: Deglitch or Debounce.

Enumerator

***LineInputFilterSelector\_Deglitch***  
***LineInputFilterSelector\_Debounce***  
***NUM\_LINEINPUTFILTERSELECTOR***

## 8.8.2.116 enum LineModeEnums

< Controls if the physical Line is used to Input or Output a signal.

Enumerator

***LineMode\_Input***  
***LineMode\_Output***  
***NUM\_LINEMODE***

## 8.8.2.117 enum LineSelectorEnums

< Selects the physical line (or pin) of the external device connector to configure

Enumerator

***LineSelector\_Line0***  
***LineSelector\_Line1***  
***LineSelector\_Line2***  
***LineSelector\_Line3***  
***NUM\_LINESELECTOR***

## 8.8.2.118 enum LineSourceEnums

< Selects which internal acquisition or I/O source signal to output on the selected line. LineMode must be Output.

## Enumerator

***LineSource\_Off***  
***LineSource\_Line0***  
***LineSource\_Line1***  
***LineSource\_Line2***  
***LineSource\_Line3***  
***LineSource\_UserOutput0***  
***LineSource\_UserOutput1***  
***LineSource\_UserOutput2***  
***LineSource\_UserOutput3***  
***LineSource\_Counter0Active***  
***LineSource\_Counter1Active***  
***LineSource\_LogicBlock0***  
***LineSource\_LogicBlock1***  
***LineSource\_ExposureActive***  
***LineSource\_FrameTriggerWait***  
***LineSource\_SerialPort0***  
***LineSource\_PPSSignal***  
***LineSource\_AllPixel***  
***LineSource\_AnyPixel***  
***NUM\_LINESOURCE***

## 8.8.2.119 enum LogicBlockLUTInputActivationEnums

< Selects the activation mode of the Logic Input Source signal.

## Enumerator

***LogicBlockLUTInputActivation\_LevelLow***  
***LogicBlockLUTInputActivation\_LevelHigh***  
***LogicBlockLUTInputActivation\_FallingEdge***  
***LogicBlockLUTInputActivation\_RisingEdge***  
***LogicBlockLUTInputActivation\_AnyEdge***  
***NUM\_LOGICBLOCKLUTINPUTACTIVATION***

## 8.8.2.120 enum LogicBlockLUTInputSelectorEnums

< Controls which LogicBlockLUT Input Source & Activation to access.

## Enumerator

***LogicBlockLUTInputSelector\_Input0***  
***LogicBlockLUTInputSelector\_Input1***  
***LogicBlockLUTInputSelector\_Input2***  
***LogicBlockLUTInputSelector\_Input3***  
***NUM\_LOGICBLOCKLUTINPUTSELECTOR***

## 8.8.2.121 enum LogicBlockLUTInputSourceEnums

< Selects the source for the input into the Logic LUT.

## Enumerator

**LogicBlockLUTInputSource\_Zero** Zero  
**LogicBlockLUTInputSource\_Line0** Line0  
**LogicBlockLUTInputSource\_Line1** Line1  
**LogicBlockLUTInputSource\_Line2** Line2  
**LogicBlockLUTInputSource\_Line3** Line3  
**LogicBlockLUTInputSource\_UserOutput0** UserOutput0  
**LogicBlockLUTInputSource\_UserOutput1** UserOutput1  
**LogicBlockLUTInputSource\_UserOutput2** UserOutput2  
**LogicBlockLUTInputSource\_UserOutput3** UserOutput3  
**LogicBlockLUTInputSource\_Counter0Start** Counter0Start  
**LogicBlockLUTInputSource\_Counter1Start** Counter1Start  
**LogicBlockLUTInputSource\_Counter0End** Counter0End  
**LogicBlockLUTInputSource\_Counter1End** Counter1End  
**LogicBlockLUTInputSource\_LogicBlock0** LogicBlock0  
**LogicBlockLUTInputSource\_LogicBlock1** LogicBlock1  
**LogicBlockLUTInputSource\_ExposureStart** ExposureStart  
**LogicBlockLUTInputSource\_ExposureEnd** ExposureEnd  
**LogicBlockLUTInputSource\_FrameTriggerWait** FrameTriggerWait  
**LogicBlockLUTInputSource\_AcquisitionActive** AcquisitionActive  
**NUM\_LOGICBLOCKLUTINPUTSOURCE**

## 8.8.2.122 enum LogicBlockLUTSelectorEnums

< Selects which LogicBlock LUT to configure

## Enumerator

**LogicBlockLUTSelector\_Value**  
**LogicBlockLUTSelector\_Enable**  
**NUM\_LOGICBLOCKLUTSELECTOR**

## 8.8.2.123 enum LogicBlockSelectorEnums

< Selects which LogicBlock to configure

## Enumerator

**LogicBlockSelector\_LogicBlock0**  
**LogicBlockSelector\_LogicBlock1**  
**NUM\_LOGICBLOCKSELECTOR**

## 8.8.2.124 enum LUTSelectorEnums

The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.

< Selects which LUT to control.

Enumerator

**LUTSelector\_LUT1** This LUT is for re-mapping pixels of all formats (mono, Bayer, red, green and blue).  
**NUM\_LUTSELECTOR**

## 8.8.2.125 enum PixelColorFilterEnums

< Type of color filter that is applied to the image. Only applies to Bayer pixel formats. All others have no color filter.

Enumerator

**PixelColorFilter\_None** No color filter.  
**PixelColorFilter\_BayerRG** Bayer Red Green filter.  
**PixelColorFilter\_BayerGB** Bayer Green Blue filter.  
**PixelColorFilter\_BayerGR** Bayer Green Red filter.  
**PixelColorFilter\_BayerBG** Bayer Blue Green filter.  
**NUM\_PIXELCOLORFILTER**

## 8.8.2.126 enum PixelFormatEnums

< Format of the pixel provided by the camera.

Enumerator

**PixelFormat\_Mono8**  
**PixelFormat\_Mono16**  
**PixelFormat\_RGB8Packed**  
**PixelFormat\_BayerGR8**  
**PixelFormat\_BayerRG8**  
**PixelFormat\_BayerGB8**  
**PixelFormat\_BayerBG8**  
**PixelFormat\_BayerGR16**  
**PixelFormat\_BayerRG16**  
**PixelFormat\_BayerGB16**  
**PixelFormat\_BayerBG16**  
**PixelFormat\_Mono12Packed**  
**PixelFormat\_BayerGR12Packed**  
**PixelFormat\_BayerRG12Packed**  
**PixelFormat\_BayerGB12Packed**  
**PixelFormat\_BayerBG12Packed**  
**PixelFormat\_YUV411Packed**



***PixelFormat\_YUV422Packed***  
***PixelFormat\_YUV444Packed***  
***PixelFormat\_Mono12p***  
***PixelFormat\_BayerGR12p***  
***PixelFormat\_BayerRG12p***  
***PixelFormat\_BayerGB12p***  
***PixelFormat\_BayerBG12p***  
***PixelFormat\_YCbCr8***  
***PixelFormat\_YCbCr422\_8***  
***PixelFormat\_YCbCr411\_8***  
***PixelFormat\_BGR8***  
***PixelFormat\_BGRa8***  
***PixelFormat\_Mono10Packed***  
***PixelFormat\_BayerGR10Packed***  
***PixelFormat\_BayerRG10Packed***  
***PixelFormat\_BayerGB10Packed***  
***PixelFormat\_BayerBG10Packed***  
***PixelFormat\_Mono10p***  
***PixelFormat\_BayerGR10p***  
***PixelFormat\_BayerRG10p***  
***PixelFormat\_BayerGB10p***  
***PixelFormat\_BayerBG10p***  
***PixelFormat\_Mono1p*** Monochrome 1-bit packed  
***PixelFormat\_Mono2p*** Monochrome 2-bit packed  
***PixelFormat\_Mono4p*** Monochrome 4-bit packed  
***PixelFormat\_Mono8s*** Monochrome 8-bit signed  
***PixelFormat\_Mono10*** Monochrome 10-bit unpacked  
***PixelFormat\_Mono12*** Monochrome 12-bit unpacked  
***PixelFormat\_Mono14*** Monochrome 14-bit unpacked  
***PixelFormat\_BayerBG10*** Bayer Blue-Green 10-bit unpacked  
***PixelFormat\_BayerBG12*** Bayer Blue-Green 12-bit unpacked  
***PixelFormat\_BayerGB10*** Bayer Green-Blue 10-bit unpacked  
***PixelFormat\_BayerGB12*** Bayer Green-Blue 12-bit unpacked  
***PixelFormat\_BayerGR10*** Bayer Green-Red 10-bit unpacked  
***PixelFormat\_BayerGR12*** Bayer Green-Red 12-bit unpacked  
***PixelFormat\_BayerRG10*** Bayer Red-Green 10-bit unpacked  
***PixelFormat\_BayerRG12*** Bayer Red-Green 12-bit unpacked  
***PixelFormat\_RGBa8*** Red-Green-Blue-alpha 8-bit  
***PixelFormat\_RGBa10*** Red-Green-Blue-alpha 10-bit unpacked  
***PixelFormat\_RGBa10p*** Red-Green-Blue-alpha 10-bit packed  
***PixelFormat\_RGBa12*** Red-Green-Blue-alpha 12-bit unpacked  
***PixelFormat\_RGBa12p*** Red-Green-Blue-alpha 12-bit packed  
***PixelFormat\_RGBa14*** Red-Green-Blue-alpha 14-bit unpacked  
***PixelFormat\_RGBa16*** Red-Green-Blue-alpha 16-bit  
***PixelFormat\_RGB8*** Red-Green-Blue 8-bit

***PixelFormat\_RGB8\_Planar*** Red-Green-Blue 8-bit planar  
***PixelFormat\_RGB10*** Red-Green-Blue 10-bit unpacked  
***PixelFormat\_RGB10\_Planar*** Red-Green-Blue 10-bit unpacked planar  
***PixelFormat\_RGB10p*** Red-Green-Blue 10-bit packed  
***PixelFormat\_RGB10p32*** Red-Green-Blue 10-bit packed into 32-bit  
***PixelFormat\_RGB12*** Red-Green-Blue 12-bit unpacked  
***PixelFormat\_RGB12\_Planar*** Red-Green-Blue 12-bit unpacked planar  
***PixelFormat\_RGB12p*** Red-Green-Blue 12-bit packed  
***PixelFormat\_RGB14*** Red-Green-Blue 14-bit unpacked  
***PixelFormat\_RGB16*** Red-Green-Blue 16-bit  
***PixelFormat\_RGB16\_Planar*** Red-Green-Blue 16-bit planar  
***PixelFormat\_RGB565p*** Red-Green-Blue 5/6/5-bit packed  
***PixelFormat\_BGRa10*** Blue-Green-Red-alpha 10-bit unpacked  
***PixelFormat\_BGRa10p*** Blue-Green-Red-alpha 10-bit packed  
***PixelFormat\_BGRa12*** Blue-Green-Red-alpha 12-bit unpacked  
***PixelFormat\_BGRa12p*** Blue-Green-Red-alpha 12-bit packed  
***PixelFormat\_BGRa14*** Blue-Green-Red-alpha 14-bit unpacked  
***PixelFormat\_BGRa16*** Blue-Green-Red-alpha 16-bit  
***PixelFormat\_BGR10*** Blue-Green-Red 10-bit unpacked  
***PixelFormat\_BGR10p*** Blue-Green-Red 10-bit packed  
***PixelFormat\_BGR12*** Blue-Green-Red 12-bit unpacked  
***PixelFormat\_BGR12p*** Blue-Green-Red 12-bit packed  
***PixelFormat\_BGR14*** Blue-Green-Red 14-bit unpacked  
***PixelFormat\_BGR16*** Blue-Green-Red 16-bit  
***PixelFormat\_BGR565p*** Blue-Green-Red 5/6/5-bit packed  
***PixelFormat\_R8*** Red 8-bit  
***PixelFormat\_R10*** Red 10-bit  
***PixelFormat\_R12*** Red 12-bit  
***PixelFormat\_R16*** Red 16-bit  
***PixelFormat\_G8*** Green 8-bit  
***PixelFormat\_G10*** Green 10-bit  
***PixelFormat\_G12*** Green 12-bit  
***PixelFormat\_G16*** Green 16-bit  
***PixelFormat\_B8*** Blue 8-bit  
***PixelFormat\_B10*** Blue 10-bit  
***PixelFormat\_B12*** Blue 12-bit  
***PixelFormat\_B16*** Blue 16-bit  
***PixelFormat\_Coord3D\_ABC8*** 3D coordinate A-B-C 8-bit  
***PixelFormat\_Coord3D\_ABC8\_Planar*** 3D coordinate A-B-C 8-bit planar  
***PixelFormat\_Coord3D\_ABC10p*** 3D coordinate A-B-C 10-bit packed  
***PixelFormat\_Coord3D\_ABC10p\_Planar*** 3D coordinate A-B-C 10-bit packed planar  
***PixelFormat\_Coord3D\_ABC12p*** 3D coordinate A-B-C 12-bit packed  
***PixelFormat\_Coord3D\_ABC12p\_Planar*** 3D coordinate A-B-C 12-bit packed planar  
***PixelFormat\_Coord3D\_ABC16*** 3D coordinate A-B-C 16-bit  
***PixelFormat\_Coord3D\_ABC16\_Planar*** 3D coordinate A-B-C 16-bit planar

***PixelFormat\_Coord3D\_ABC32f*** 3D coordinate A-B-C 32-bit floating point  
***PixelFormat\_Coord3D\_ABC32f\_Planar*** 3D coordinate A-B-C 32-bit floating point planar  
***PixelFormat\_Coord3D\_AC8*** 3D coordinate A-C 8-bit  
***PixelFormat\_Coord3D\_AC8\_Planar*** 3D coordinate A-C 8-bit planar  
***PixelFormat\_Coord3D\_AC10p*** 3D coordinate A-C 10-bit packed  
***PixelFormat\_Coord3D\_AC10p\_Planar*** 3D coordinate A-C 10-bit packed planar  
***PixelFormat\_Coord3D\_AC12p*** 3D coordinate A-C 12-bit packed  
***PixelFormat\_Coord3D\_AC12p\_Planar*** 3D coordinate A-C 12-bit packed planar  
***PixelFormat\_Coord3D\_AC16*** 3D coordinate A-C 16-bit  
***PixelFormat\_Coord3D\_AC16\_Planar*** 3D coordinate A-C 16-bit planar  
***PixelFormat\_Coord3D\_AC32f*** 3D coordinate A-C 32-bit floating point  
***PixelFormat\_Coord3D\_AC32f\_Planar*** 3D coordinate A-C 32-bit floating point planar  
***PixelFormat\_Coord3D\_A8*** 3D coordinate A 8-bit  
***PixelFormat\_Coord3D\_A10p*** 3D coordinate A 10-bit packed  
***PixelFormat\_Coord3D\_A12p*** 3D coordinate A 12-bit packed  
***PixelFormat\_Coord3D\_A16*** 3D coordinate A 16-bit  
***PixelFormat\_Coord3D\_A32f*** 3D coordinate A 32-bit floating point  
***PixelFormat\_Coord3D\_B8*** 3D coordinate B 8-bit  
***PixelFormat\_Coord3D\_B10p*** 3D coordinate B 10-bit packed  
***PixelFormat\_Coord3D\_B12p*** 3D coordinate B 12-bit packed  
***PixelFormat\_Coord3D\_B16*** 3D coordinate B 16-bit  
***PixelFormat\_Coord3D\_B32f*** 3D coordinate B 32-bit floating point  
***PixelFormat\_Coord3D\_C8*** 3D coordinate C 8-bit  
***PixelFormat\_Coord3D\_C10p*** 3D coordinate C 10-bit packed  
***PixelFormat\_Coord3D\_C12p*** 3D coordinate C 12-bit packed  
***PixelFormat\_Coord3D\_C16*** 3D coordinate C 16-bit  
***PixelFormat\_Coord3D\_C32f*** 3D coordinate C 32-bit floating point  
***PixelFormat\_Confidence1*** Confidence 1-bit unpacked  
***PixelFormat\_Confidence1p*** Confidence 1-bit packed  
***PixelFormat\_Confidence8*** Confidence 8-bit  
***PixelFormat\_Confidence16*** Confidence 16-bit  
***PixelFormat\_Confidence32f*** Confidence 32-bit floating point  
***PixelFormat\_BiColorBGRG8*** Bi-color Blue/Green - Red/Green 8-bit  
***PixelFormat\_BiColorBGRG10*** Bi-color Blue/Green - Red/Green 10-bit unpacked  
***PixelFormat\_BiColorBGRG10p*** Bi-color Blue/Green - Red/Green 10-bit packed  
***PixelFormat\_BiColorBGRG12*** Bi-color Blue/Green - Red/Green 12-bit unpacked  
***PixelFormat\_BiColorBGRG12p*** Bi-color Blue/Green - Red/Green 12-bit packed  
***PixelFormat\_BiColorRGBG8*** Bi-color Red/Green - Blue/Green 8-bit  
***PixelFormat\_BiColorRGBG10*** Bi-color Red/Green - Blue/Green 10-bit unpacked  
***PixelFormat\_BiColorRGBG10p*** Bi-color Red/Green - Blue/Green 10-bit packed  
***PixelFormat\_BiColorRGBG12*** Bi-color Red/Green - Blue/Green 12-bit unpacked  
***PixelFormat\_BiColorRGBG12p*** Bi-color Red/Green - Blue/Green 12-bit packed  
***PixelFormat\_SCF1WBWG8*** Sparse Color Filter #1 White-Blue-White-Green 8-bit  
***PixelFormat\_SCF1WBWG10*** Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked  
***PixelFormat\_SCF1WBWG10p*** Sparse Color Filter #1 White-Blue-White-Green 10-bit packed

**PixelFormat\_SCF1WBWG12** Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked  
**PixelFormat\_SCF1WBWG12p** Sparse Color Filter #1 White-Blue-White-Green 12-bit packed  
**PixelFormat\_SCF1WBWG14** Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked  
**PixelFormat\_SCF1WBWG16** Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked  
**PixelFormat\_SCF1WGWB8** Sparse Color Filter #1 White-Green-White-Blue 8-bit  
**PixelFormat\_SCF1WGWB10** Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked  
**PixelFormat\_SCF1WGWB10p** Sparse Color Filter #1 White-Green-White-Blue 10-bit packed  
**PixelFormat\_SCF1WGWB12** Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked  
**PixelFormat\_SCF1WGWB12p** Sparse Color Filter #1 White-Green-White-Blue 12-bit packed  
**PixelFormat\_SCF1WGWB14** Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked  
**PixelFormat\_SCF1WGWB16** Sparse Color Filter #1 White-Green-White-Blue 16-bit  
**PixelFormat\_SCF1WGWR8** Sparse Color Filter #1 White-Green-White-Red 8-bit  
**PixelFormat\_SCF1WGWR10** Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked  
**PixelFormat\_SCF1WGWR10p** Sparse Color Filter #1 White-Green-White-Red 10-bit packed  
**PixelFormat\_SCF1WGWR12** Sparse Color Filter #1 White-Green-White-Red 12-bit unpacked  
**PixelFormat\_SCF1WGWR12p** Sparse Color Filter #1 White-Green-White-Red 12-bit packed  
**PixelFormat\_SCF1WGWR14** Sparse Color Filter #1 White-Green-White-Red 14-bit unpacked  
**PixelFormat\_SCF1WGWR16** Sparse Color Filter #1 White-Green-White-Red 16-bit  
**PixelFormat\_SCF1WRWG8** Sparse Color Filter #1 White-Red-White-Green 8-bit  
**PixelFormat\_SCF1WRWG10** Sparse Color Filter #1 White-Red-White-Green 10-bit unpacked  
**PixelFormat\_SCF1WRWG10p** Sparse Color Filter #1 White-Red-White-Green 10-bit packed  
**PixelFormat\_SCF1WRWG12** Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked  
**PixelFormat\_SCF1WRWG12p** Sparse Color Filter #1 White-Red-White-Green 12-bit packed  
**PixelFormat\_SCF1WRWG14** Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked  
**PixelFormat\_SCF1WRWG16** Sparse Color Filter #1 White-Red-White-Green 16-bit  
**PixelFormat\_YCbCr8\_CbYCr** YCbCr 4:4:4 8-bit  
**PixelFormat\_YCbCr10\_CbYCr** YCbCr 4:4:4 10-bit unpacked  
**PixelFormat\_YCbCr10p\_CbYCr** YCbCr 4:4:4 10-bit packed  
**PixelFormat\_YCbCr12\_CbYCr** YCbCr 4:4:4 12-bit unpacked  
**PixelFormat\_YCbCr12p\_CbYCr** YCbCr 4:4:4 12-bit packed  
**PixelFormat\_YCbCr411\_8\_CbYYCrYY** YCbCr 4:1:1 8-bit  
**PixelFormat\_YCbCr422\_8\_CbYCrY** YCbCr 4:2:2 8-bit  
**PixelFormat\_YCbCr422\_10** YCbCr 4:2:2 10-bit unpacked  
**PixelFormat\_YCbCr422\_10\_CbYCrY** YCbCr 4:2:2 10-bit unpacked  
**PixelFormat\_YCbCr422\_10p** YCbCr 4:2:2 10-bit packed  
**PixelFormat\_YCbCr422\_10p\_CbYCrY** YCbCr 4:2:2 10-bit packed  
**PixelFormat\_YCbCr422\_12** YCbCr 4:2:2 12-bit unpacked  
**PixelFormat\_YCbCr422\_12\_CbYCrY** YCbCr 4:2:2 12-bit unpacked  
**PixelFormat\_YCbCr422\_12p** YCbCr 4:2:2 12-bit packed  
**PixelFormat\_YCbCr422\_12p\_CbYCrY** YCbCr 4:2:2 12-bit packed  
**PixelFormat\_YCbCr601\_8\_CbYCr** YCbCr 4:4:4 8-bit BT.601  
**PixelFormat\_YCbCr601\_10\_CbYCr** YCbCr 4:4:4 10-bit unpacked BT.601  
**PixelFormat\_YCbCr601\_10p\_CbYCr** YCbCr 4:4:4 10-bit packed BT.601  
**PixelFormat\_YCbCr601\_12\_CbYCr** YCbCr 4:4:4 12-bit unpacked BT.601  
**PixelFormat\_YCbCr601\_12p\_CbYCr** YCbCr 4:4:4 12-bit packed BT.601

**PixelFormat\_YCbCr601\_411\_8\_CbYYCrYY** YCbCr 4:1:1 8-bit BT.601  
**PixelFormat\_YCbCr601\_422\_8** YCbCr 4:2:2 8-bit BT.601  
**PixelFormat\_YCbCr601\_422\_8\_CbYCrY** YCbCr 4:2:2 8-bit BT.601  
**PixelFormat\_YCbCr601\_422\_10** YCbCr 4:2:2 10-bit unpacked BT.601  
**PixelFormat\_YCbCr601\_422\_10\_CbYCrY** YCbCr 4:2:2 10-bit unpacked BT.601  
**PixelFormat\_YCbCr601\_422\_10p** YCbCr 4:2:2 10-bit packed BT.601  
**PixelFormat\_YCbCr601\_422\_10p\_CbYCrY** YCbCr 4:2:2 10-bit packed BT.601  
**PixelFormat\_YCbCr601\_422\_12** YCbCr 4:2:2 12-bit unpacked BT.601  
**PixelFormat\_YCbCr601\_422\_12\_CbYCrY** YCbCr 4:2:2 12-bit unpacked BT.601  
**PixelFormat\_YCbCr601\_422\_12p** YCbCr 4:2:2 12-bit packed BT.601  
**PixelFormat\_YCbCr601\_422\_12p\_CbYCrY** YCbCr 4:2:2 12-bit packed BT.601  
**PixelFormat\_YCbCr709\_8\_CbYCr** YCbCr 4:4:4 8-bit BT.709  
**PixelFormat\_YCbCr709\_10\_CbYCr** YCbCr 4:4:4 10-bit unpacked BT.709  
**PixelFormat\_YCbCr709\_10p\_CbYCr** YCbCr 4:4:4 10-bit packed BT.709  
**PixelFormat\_YCbCr709\_12\_CbYCr** YCbCr 4:4:4 12-bit unpacked BT.709  
**PixelFormat\_YCbCr709\_12p\_CbYCr** YCbCr 4:4:4 12-bit packed BT.709  
**PixelFormat\_YCbCr709\_411\_8\_CbYYCrYY** YCbCr 4:1:1 8-bit BT.709  
**PixelFormat\_YCbCr709\_422\_8** YCbCr 4:2:2 8-bit BT.709  
**PixelFormat\_YCbCr709\_422\_8\_CbYCrY** YCbCr 4:2:2 8-bit BT.709  
**PixelFormat\_YCbCr709\_422\_10** YCbCr 4:2:2 10-bit unpacked BT.709  
**PixelFormat\_YCbCr709\_422\_10\_CbYCrY** YCbCr 4:2:2 10-bit unpacked BT.709  
**PixelFormat\_YCbCr709\_422\_10p** YCbCr 4:2:2 10-bit packed BT.709  
**PixelFormat\_YCbCr709\_422\_10p\_CbYCrY** YCbCr 4:2:2 10-bit packed BT.709  
**PixelFormat\_YCbCr709\_422\_12** YCbCr 4:2:2 12-bit unpacked BT.709  
**PixelFormat\_YCbCr709\_422\_12\_CbYCrY** YCbCr 4:2:2 12-bit unpacked BT.709  
**PixelFormat\_YCbCr709\_422\_12p** YCbCr 4:2:2 12-bit packed BT.709  
**PixelFormat\_YCbCr709\_422\_12p\_CbYCrY** YCbCr 4:2:2 12-bit packed BT.709  
**PixelFormat\_YUV8\_UYV** YUV 4:4:4 8-bit  
**PixelFormat\_YUV411\_8\_UYYVYY** YUV 4:1:1 8-bit  
**PixelFormat\_YUV422\_8** YUV 4:2:2 8-bit  
**PixelFormat\_YUV422\_8\_UYVY** YUV 4:2:2 8-bit  
**PixelFormat\_Polarized8** Monochrome Polarized 8-bit  
**PixelFormat\_Polarized10p** Monochrome Polarized 10-bit packed  
**PixelFormat\_Polarized12p** Monochrome Polarized 12-bit packed  
**PixelFormat\_Polarized16** Monochrome Polarized 16-bit  
**PixelFormat\_BayerRGPolarized8** Polarized Bayer Red Green filter 8-bit  
**PixelFormat\_BayerRGPolarized10p** Polarized Bayer Red Green filter 10-bit packed  
**PixelFormat\_BayerRGPolarized12p** Polarized Bayer Red Green filter 12-bit packed  
**PixelFormat\_BayerRGPolarized16** Polarized Bayer Red Green filter 16-bit  
**PixelFormat\_Raw16** Raw 16 bit.  
**PixelFormat\_Raw8** Raw bit.  
**PixelFormat\_R12\_Jpeg** Red 12-bit JPEG.  
**PixelFormat\_GR12\_Jpeg** Green Red 12-bit JPEG.  
**PixelFormat\_GB12\_Jpeg** Green Blue 12-bit JPEG.  
**PixelFormat\_B12\_Jpeg** Blue 12-bit packed JPEG.  
**UNKNOWN\_PIXELFORMAT**  
**NUM\_PIXELFORMAT**

## 8.8.2.127 enum PixelFormatInfoSelectorEnums

< Select the pixel format for which the information will be returned.

## Enumerator

**PixelFormatInfoSelector\_Mono1p** Monochrome 1-bit packed  
**PixelFormatInfoSelector\_Mono2p** Monochrome 2-bit packed  
**PixelFormatInfoSelector\_Mono4p** Monochrome 4-bit packed  
**PixelFormatInfoSelector\_Mono8** Monochrome 8-bit  
**PixelFormatInfoSelector\_Mono8s** Monochrome 8-bit signed  
**PixelFormatInfoSelector\_Mono10** Monochrome 10-bit unpacked  
**PixelFormatInfoSelector\_Mono10p** Monochrome 10-bit packed  
**PixelFormatInfoSelector\_Mono12** Monochrome 12-bit unpacked  
**PixelFormatInfoSelector\_Mono12p** Monochrome 12-bit packed  
**PixelFormatInfoSelector\_Mono14** Monochrome 14-bit unpacked  
**PixelFormatInfoSelector\_Mono16** Monochrome 16-bit  
**PixelFormatInfoSelector\_BayerBG8** Bayer Blue-Green 8-bit  
**PixelFormatInfoSelector\_BayerBG10** Bayer Blue-Green 10-bit unpacked  
**PixelFormatInfoSelector\_BayerBG10p** Bayer Blue-Green 10-bit packed  
**PixelFormatInfoSelector\_BayerBG12** Bayer Blue-Green 12-bit unpacked  
**PixelFormatInfoSelector\_BayerBG12p** Bayer Blue-Green 12-bit packed  
**PixelFormatInfoSelector\_BayerBG16** Bayer Blue-Green 16-bit  
**PixelFormatInfoSelector\_BayerGB8** Bayer Green-Blue 8-bit  
**PixelFormatInfoSelector\_BayerGB10** Bayer Green-Blue 10-bit unpacked  
**PixelFormatInfoSelector\_BayerGB10p** Bayer Green-Blue 10-bit packed  
**PixelFormatInfoSelector\_BayerGB12** Bayer Green-Blue 12-bit unpacked  
**PixelFormatInfoSelector\_BayerGB12p** Bayer Green-Blue 12-bit packed  
**PixelFormatInfoSelector\_BayerGB16** Bayer Green-Blue 16-bit  
**PixelFormatInfoSelector\_BayerGR8** Bayer Green-Red 8-bit  
**PixelFormatInfoSelector\_BayerGR10** Bayer Green-Red 10-bit unpacked  
**PixelFormatInfoSelector\_BayerGR10p** Bayer Green-Red 10-bit packed  
**PixelFormatInfoSelector\_BayerGR12** Bayer Green-Red 12-bit unpacked  
**PixelFormatInfoSelector\_BayerGR12p** Bayer Green-Red 12-bit packed  
**PixelFormatInfoSelector\_BayerGR16** Bayer Green-Red 16-bit  
**PixelFormatInfoSelector\_BayerRG8** Bayer Red-Green 8-bit  
**PixelFormatInfoSelector\_BayerRG10** Bayer Red-Green 10-bit unpacked  
**PixelFormatInfoSelector\_BayerRG10p** Bayer Red-Green 10-bit packed  
**PixelFormatInfoSelector\_BayerRG12** Bayer Red-Green 12-bit unpacked  
**PixelFormatInfoSelector\_BayerRG12p** Bayer Red-Green 12-bit packed  
**PixelFormatInfoSelector\_BayerRG16** Bayer Red-Green 16-bit  
**PixelFormatInfoSelector\_RGBa8** Red-Green-Blue-alpha 8-bit  
**PixelFormatInfoSelector\_RGBa10** Red-Green-Blue-alpha 10-bit unpacked  
**PixelFormatInfoSelector\_RGBa10p** Red-Green-Blue-alpha 10-bit packed  
**PixelFormatInfoSelector\_RGBa12** Red-Green-Blue-alpha 12-bit unpacked  
**PixelFormatInfoSelector\_RGBa12p** Red-Green-Blue-alpha 12-bit packed

***PixelFormatInfoSelector\_RGBA14*** Red-Green-Blue-alpha 14-bit unpacked  
***PixelFormatInfoSelector\_RGBA16*** Red-Green-Blue-alpha 16-bit  
***PixelFormatInfoSelector\_RGB8*** Red-Green-Blue 8-bit  
***PixelFormatInfoSelector\_RGB8\_Planar*** Red-Green-Blue 8-bit planar  
***PixelFormatInfoSelector\_RGB10*** Red-Green-Blue 10-bit unpacked  
***PixelFormatInfoSelector\_RGB10\_Planar*** Red-Green-Blue 10-bit unpacked planar  
***PixelFormatInfoSelector\_RGB10p*** Red-Green-Blue 10-bit packed  
***PixelFormatInfoSelector\_RGB10p32*** Red-Green-Blue 10-bit packed into 32-bit  
***PixelFormatInfoSelector\_RGB12*** Red-Green-Blue 12-bit unpacked  
***PixelFormatInfoSelector\_RGB12\_Planar*** Red-Green-Blue 12-bit unpacked planar  
***PixelFormatInfoSelector\_RGB12p*** Red-Green-Blue 12-bit packed  
***PixelFormatInfoSelector\_RGB14*** Red-Green-Blue 14-bit unpacked  
***PixelFormatInfoSelector\_RGB16*** Red-Green-Blue 16-bit  
***PixelFormatInfoSelector\_RGB16\_Planar*** Red-Green-Blue 16-bit planar  
***PixelFormatInfoSelector\_RGB565p*** Red-Green-Blue 5/6/5-bit packed  
***PixelFormatInfoSelector\_BGRa8*** Blue-Green-Red-alpha 8-bit  
***PixelFormatInfoSelector\_BGRa10*** Blue-Green-Red-alpha 10-bit unpacked  
***PixelFormatInfoSelector\_BGRa10p*** Blue-Green-Red-alpha 10-bit packed  
***PixelFormatInfoSelector\_BGRa12*** Blue-Green-Red-alpha 12-bit unpacked  
***PixelFormatInfoSelector\_BGRa12p*** Blue-Green-Red-alpha 12-bit packed  
***PixelFormatInfoSelector\_BGRa14*** Blue-Green-Red-alpha 14-bit unpacked  
***PixelFormatInfoSelector\_BGRa16*** Blue-Green-Red-alpha 16-bit  
***PixelFormatInfoSelector\_BGR8*** Blue-Green-Red 8-bit  
***PixelFormatInfoSelector\_BGR10*** Blue-Green-Red 10-bit unpacked  
***PixelFormatInfoSelector\_BGR10p*** Blue-Green-Red 10-bit packed  
***PixelFormatInfoSelector\_BGR12*** Blue-Green-Red 12-bit unpacked  
***PixelFormatInfoSelector\_BGR12p*** Blue-Green-Red 12-bit packed  
***PixelFormatInfoSelector\_BGR14*** Blue-Green-Red 14-bit unpacked  
***PixelFormatInfoSelector\_BGR16*** Blue-Green-Red 16-bit  
***PixelFormatInfoSelector\_BGR565p*** Blue-Green-Red 5/6/5-bit packed  
***PixelFormatInfoSelector\_R8*** Red 8-bit  
***PixelFormatInfoSelector\_R10*** Red 10-bit  
***PixelFormatInfoSelector\_R12*** Red 12-bit  
***PixelFormatInfoSelector\_R16*** Red 16-bit  
***PixelFormatInfoSelector\_G8*** Green 8-bit  
***PixelFormatInfoSelector\_G10*** Green 10-bit  
***PixelFormatInfoSelector\_G12*** Green 12-bit  
***PixelFormatInfoSelector\_G16*** Green 16-bit  
***PixelFormatInfoSelector\_B8*** Blue 8-bit  
***PixelFormatInfoSelector\_B10*** Blue 10-bit  
***PixelFormatInfoSelector\_B12*** Blue 12-bit  
***PixelFormatInfoSelector\_B16*** Blue 16-bit  
***PixelFormatInfoSelector\_Coord3D\_ABC8*** 3D coordinate A-B-C 8-bit  
***PixelFormatInfoSelector\_Coord3D\_ABC8\_Planar*** 3D coordinate A-B-C 8-bit planar  
***PixelFormatInfoSelector\_Coord3D\_ABC10p*** 3D coordinate A-B-C 10-bit packed

***PixelFormatInfoSelector\_Coord3D\_ABC10p\_Planar*** 3D coordinate A-B-C 10-bit packed planar  
***PixelFormatInfoSelector\_Coord3D\_ABC12p*** 3D coordinate A-B-C 12-bit packed  
***PixelFormatInfoSelector\_Coord3D\_ABC12p\_Planar*** 3D coordinate A-B-C 12-bit packed planar  
***PixelFormatInfoSelector\_Coord3D\_ABC16*** 3D coordinate A-B-C 16-bit  
***PixelFormatInfoSelector\_Coord3D\_ABC16\_Planar*** 3D coordinate A-B-C 16-bit planar  
***PixelFormatInfoSelector\_Coord3D\_ABC32f*** 3D coordinate A-B-C 32-bit floating point  
***PixelFormatInfoSelector\_Coord3D\_ABC32f\_Planar*** 3D coordinate A-B-C 32-bit floating point planar  
***PixelFormatInfoSelector\_Coord3D\_AC8*** 3D coordinate A-C 8-bit  
***PixelFormatInfoSelector\_Coord3D\_AC8\_Planar*** 3D coordinate A-C 8-bit planar  
***PixelFormatInfoSelector\_Coord3D\_AC10p*** 3D coordinate A-C 10-bit packed  
***PixelFormatInfoSelector\_Coord3D\_AC10p\_Planar*** 3D coordinate A-C 10-bit packed planar  
***PixelFormatInfoSelector\_Coord3D\_AC12p*** 3D coordinate A-C 12-bit packed  
***PixelFormatInfoSelector\_Coord3D\_AC12p\_Planar*** 3D coordinate A-C 12-bit packed planar  
***PixelFormatInfoSelector\_Coord3D\_AC16*** 3D coordinate A-C 16-bit  
***PixelFormatInfoSelector\_Coord3D\_AC16\_Planar*** 3D coordinate A-C 16-bit planar  
***PixelFormatInfoSelector\_Coord3D\_AC32f*** 3D coordinate A-C 32-bit floating point  
***PixelFormatInfoSelector\_Coord3D\_AC32f\_Planar*** 3D coordinate A-C 32-bit floating point planar  
***PixelFormatInfoSelector\_Coord3D\_A8*** 3D coordinate A 8-bit  
***PixelFormatInfoSelector\_Coord3D\_A10p*** 3D coordinate A 10-bit packed  
***PixelFormatInfoSelector\_Coord3D\_A12p*** 3D coordinate A 12-bit packed  
***PixelFormatInfoSelector\_Coord3D\_A16*** 3D coordinate A 16-bit  
***PixelFormatInfoSelector\_Coord3D\_A32f*** 3D coordinate A 32-bit floating point  
***PixelFormatInfoSelector\_Coord3D\_B8*** 3D coordinate B 8-bit  
***PixelFormatInfoSelector\_Coord3D\_B10p*** 3D coordinate B 10-bit packed  
***PixelFormatInfoSelector\_Coord3D\_B12p*** 3D coordinate B 12-bit packed  
***PixelFormatInfoSelector\_Coord3D\_B16*** 3D coordinate B 16-bit  
***PixelFormatInfoSelector\_Coord3D\_B32f*** 3D coordinate B 32-bit floating point  
***PixelFormatInfoSelector\_Coord3D\_C8*** 3D coordinate C 8-bit  
***PixelFormatInfoSelector\_Coord3D\_C10p*** 3D coordinate C 10-bit packed  
***PixelFormatInfoSelector\_Coord3D\_C12p*** 3D coordinate C 12-bit packed  
***PixelFormatInfoSelector\_Coord3D\_C16*** 3D coordinate C 16-bit  
***PixelFormatInfoSelector\_Coord3D\_C32f*** 3D coordinate C 32-bit floating point  
***PixelFormatInfoSelector\_Confidence1*** Confidence 1-bit unpacked  
***PixelFormatInfoSelector\_Confidence1p*** Confidence 1-bit packed  
***PixelFormatInfoSelector\_Confidence8*** Confidence 8-bit  
***PixelFormatInfoSelector\_Confidence16*** Confidence 16-bit  
***PixelFormatInfoSelector\_Confidence32f*** Confidence 32-bit floating point  
***PixelFormatInfoSelector\_BiColorBGRG8*** Bi-color Blue/Green - Red/Green 8-bit  
***PixelFormatInfoSelector\_BiColorBGRG10*** Bi-color Blue/Green - Red/Green 10-bit unpacked  
***PixelFormatInfoSelector\_BiColorBGRG10p*** Bi-color Blue/Green - Red/Green 10-bit packed  
***PixelFormatInfoSelector\_BiColorBGRG12*** Bi-color Blue/Green - Red/Green 12-bit unpacked  
***PixelFormatInfoSelector\_BiColorBGRG12p*** Bi-color Blue/Green - Red/Green 12-bit packed  
***PixelFormatInfoSelector\_BiColorRGBG8*** Bi-color Red/Green - Blue/Green 8-bit  
***PixelFormatInfoSelector\_BiColorRGBG10*** Bi-color Red/Green - Blue/Green 10-bit unpacked  
***PixelFormatInfoSelector\_BiColorRGBG10p*** Bi-color Red/Green - Blue/Green 10-bit packed



***PixelFormatInfoSelector\_BiColorRGBG12*** Bi-color Red/Green - Blue/Green 12-bit unpacked  
***PixelFormatInfoSelector\_BiColorRGBG12p*** Bi-color Red/Green - Blue/Green 12-bit packed  
***PixelFormatInfoSelector\_SCF1WBWG8*** Sparse Color Filter #1 White-Blue-White-Green 8-bit  
***PixelFormatInfoSelector\_SCF1WBWG10*** Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WBWG10p*** Sparse Color Filter #1 White-Blue-White-Green 10-bit packed  
***PixelFormatInfoSelector\_SCF1WBWG12*** Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WBWG12p*** Sparse Color Filter #1 White-Blue-White-Green 12-bit packed  
***PixelFormatInfoSelector\_SCF1WBWG14*** Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WBWG16*** Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WGWB8*** Sparse Color Filter #1 White-Green-White-Blue 8-bit  
***PixelFormatInfoSelector\_SCF1WGWB10*** Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WGWB10p*** Sparse Color Filter #1 White-Green-White-Blue 10-bit packed  
***PixelFormatInfoSelector\_SCF1WGWB12*** Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WGWB12p*** Sparse Color Filter #1 White-Green-White-Blue 12-bit packed  
***PixelFormatInfoSelector\_SCF1WGWB14*** Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WGWB16*** Sparse Color Filter #1 White-Green-White-Blue 16-bit  
***PixelFormatInfoSelector\_SCF1WGWR8*** Sparse Color Filter #1 White-Green-White-Red 8-bit  
***PixelFormatInfoSelector\_SCF1WGWR10*** Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WGWR10p*** Sparse Color Filter #1 White-Green-White-Red 10-bit packed  
***PixelFormatInfoSelector\_SCF1WGWR12*** Sparse Color Filter #1 White-Green-White-Red 12-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WGWR12p*** Sparse Color Filter #1 White-Green-White-Red 12-bit packed  
***PixelFormatInfoSelector\_SCF1WGWR14*** Sparse Color Filter #1 White-Green-White-Red 14-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WGWR16*** Sparse Color Filter #1 White-Green-White-Red 16-bit  
***PixelFormatInfoSelector\_SCF1WRWG8*** Sparse Color Filter #1 White-Red-White-Green 8-bit  
***PixelFormatInfoSelector\_SCF1WRWG10*** Sparse Color Filter #1 White-Red-White-Green 10-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WRWG10p*** Sparse Color Filter #1 White-Red-White-Green 10-bit packed  
***PixelFormatInfoSelector\_SCF1WRWG12*** Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WRWG12p*** Sparse Color Filter #1 White-Red-White-Green 12-bit packed  
***PixelFormatInfoSelector\_SCF1WRWG14*** Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked  
  
***PixelFormatInfoSelector\_SCF1WRWG16*** Sparse Color Filter #1 White-Red-White-Green 16-bit  
***PixelFormatInfoSelector\_YCbCr8*** YCbCr 4:4:4 8-bit  
***PixelFormatInfoSelector\_YCbCr8\_CbYCr*** YCbCr 4:4:4 8-bit  
***PixelFormatInfoSelector\_YCbCr10\_CbYCr*** YCbCr 4:4:4 10-bit unpacked  
***PixelFormatInfoSelector\_YCbCr10p\_CbYCr*** YCbCr 4:4:4 10-bit packed  
***PixelFormatInfoSelector\_YCbCr12\_CbYCr*** YCbCr 4:4:4 12-bit unpacked

*PixelFormatInfoSelector\_YCbCr12p\_CbYCr* YCbCr 4:4:4 12-bit packed  
*PixelFormatInfoSelector\_YCbCr411\_8* YCbCr 4:1:1 8-bit  
*PixelFormatInfoSelector\_YCbCr411\_8\_CbYYCrYY* YCbCr 4:1:1 8-bit  
*PixelFormatInfoSelector\_YCbCr422\_8* YCbCr 4:2:2 8-bit  
*PixelFormatInfoSelector\_YCbCr422\_8\_CbYCrY* YCbCr 4:2:2 8-bit  
*PixelFormatInfoSelector\_YCbCr422\_10* YCbCr 4:2:2 10-bit unpacked  
*PixelFormatInfoSelector\_YCbCr422\_10\_CbYCrY* YCbCr 4:2:2 10-bit unpacked  
*PixelFormatInfoSelector\_YCbCr422\_10p* YCbCr 4:2:2 10-bit packed  
*PixelFormatInfoSelector\_YCbCr422\_10p\_CbYCrY* YCbCr 4:2:2 10-bit packed  
*PixelFormatInfoSelector\_YCbCr422\_12* YCbCr 4:2:2 12-bit unpacked  
*PixelFormatInfoSelector\_YCbCr422\_12\_CbYCrY* YCbCr 4:2:2 12-bit unpacked  
*PixelFormatInfoSelector\_YCbCr422\_12p* YCbCr 4:2:2 12-bit packed  
*PixelFormatInfoSelector\_YCbCr422\_12p\_CbYCrY* YCbCr 4:2:2 12-bit packed  
*PixelFormatInfoSelector\_YCbCr601\_8\_CbYCr* YCbCr 4:4:4 8-bit BT.601  
*PixelFormatInfoSelector\_YCbCr601\_10\_CbYCr* YCbCr 4:4:4 10-bit unpacked BT.601  
*PixelFormatInfoSelector\_YCbCr601\_10p\_CbYCr* YCbCr 4:4:4 10-bit packed BT.601  
*PixelFormatInfoSelector\_YCbCr601\_12\_CbYCr* YCbCr 4:4:4 12-bit unpacked BT.601  
*PixelFormatInfoSelector\_YCbCr601\_12p\_CbYCr* YCbCr 4:4:4 12-bit packed BT.601  
*PixelFormatInfoSelector\_YCbCr601\_411\_8\_CbYYCrYY* YCbCr 4:1:1 8-bit BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_8* YCbCr 4:2:2 8-bit BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_8\_CbYCrY* YCbCr 4:2:2 8-bit BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_10* YCbCr 4:2:2 10-bit unpacked BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_10\_CbYCrY* YCbCr 4:2:2 10-bit unpacked BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_10p* YCbCr 4:2:2 10-bit packed BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_10p\_CbYCrY* YCbCr 4:2:2 10-bit packed BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_12* YCbCr 4:2:2 12-bit unpacked BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_12\_CbYCrY* YCbCr 4:2:2 12-bit unpacked BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_12p* YCbCr 4:2:2 12-bit packed BT.601  
*PixelFormatInfoSelector\_YCbCr601\_422\_12p\_CbYCrY* YCbCr 4:2:2 12-bit packed BT.601  
*PixelFormatInfoSelector\_YCbCr709\_8\_CbYCr* YCbCr 4:4:4 8-bit BT.709  
*PixelFormatInfoSelector\_YCbCr709\_10\_CbYCr* YCbCr 4:4:4 10-bit unpacked BT.709  
*PixelFormatInfoSelector\_YCbCr709\_10p\_CbYCr* YCbCr 4:4:4 10-bit packed BT.709  
*PixelFormatInfoSelector\_YCbCr709\_12\_CbYCr* YCbCr 4:4:4 12-bit unpacked BT.709  
*PixelFormatInfoSelector\_YCbCr709\_12p\_CbYCr* YCbCr 4:4:4 12-bit packed BT.709  
*PixelFormatInfoSelector\_YCbCr709\_411\_8\_CbYYCrYY* YCbCr 4:1:1 8-bit BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_8* YCbCr 4:2:2 8-bit BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_8\_CbYCrY* YCbCr 4:2:2 8-bit BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_10* YCbCr 4:2:2 10-bit unpacked BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_10\_CbYCrY* YCbCr 4:2:2 10-bit unpacked BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_10p* YCbCr 4:2:2 10-bit packed BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_10p\_CbYCrY* YCbCr 4:2:2 10-bit packed BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_12* YCbCr 4:2:2 12-bit unpacked BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_12\_CbYCrY* YCbCr 4:2:2 12-bit unpacked BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_12p* YCbCr 4:2:2 12-bit packed BT.709  
*PixelFormatInfoSelector\_YCbCr709\_422\_12p\_CbYCrY* YCbCr 4:2:2 12-bit packed BT.709

***PixelFormatInfoSelector\_YUV8\_UYV*** YUV 4:4:4 8-bit  
***PixelFormatInfoSelector\_YUV411\_8\_UYYVYY*** YUV 4:1:1 8-bit  
***PixelFormatInfoSelector\_YUV422\_8*** YUV 4:2:2 8-bit  
***PixelFormatInfoSelector\_YUV422\_8\_UYVY*** YUV 4:2:2 8-bit  
***PixelFormatInfoSelector\_Polarized8*** Monochrome Polarized 8-bit  
***PixelFormatInfoSelector\_Polarized10p*** Monochrome Polarized 10-bit packed  
***PixelFormatInfoSelector\_Polarized12p*** Monochrome Polarized 12-bit packed  
***PixelFormatInfoSelector\_Polarized16*** Monochrome Polarized 16-bit  
***PixelFormatInfoSelector\_BayerRGPolarized8*** Polarized Bayer Red Green filter 8-bit  
***PixelFormatInfoSelector\_BayerRGPolarized10p*** Polarized Bayer Red Green filter 10-bit packed  
***PixelFormatInfoSelector\_BayerRGPolarized12p*** Polarized Bayer Red Green filter 12-bit packed  
***PixelFormatInfoSelector\_BayerRGPolarized16*** Polarized Bayer Red Green filter 16-bit  
***NUM\_PIXELFORMATINFOSELECTOR***

#### 8.8.2.128 enum PixelSizeEnums

< Total size in bits of a pixel of the image.

##### Enumerator

***PixelSize\_Bpp1*** 1 bit per pixel.  
***PixelSize\_Bpp2*** 2 bits per pixel.  
***PixelSize\_Bpp4*** 4 bits per pixel.  
***PixelSize\_Bpp8*** 8 bits per pixel.  
***PixelSize\_Bpp10*** 10 bits per pixel.  
***PixelSize\_Bpp12*** 12 bits per pixel.  
***PixelSize\_Bpp14*** 14 bits per pixel.  
***PixelSize\_Bpp16*** 16 bits per pixel.  
***PixelSize\_Bpp20*** 20 bits per pixel.  
***PixelSize\_Bpp24*** 24 bits per pixel.  
***PixelSize\_Bpp30*** 30 bits per pixel.  
***PixelSize\_Bpp32*** 32 bits per pixel.  
***PixelSize\_Bpp36*** 36 bits per pixel.  
***PixelSize\_Bpp48*** 48 bits per pixel.  
***PixelSize\_Bpp64*** 64 bits per pixel.  
***PixelSize\_Bpp96*** 96 bits per pixel.  
***NUM\_PIXELSIZE***

#### 8.8.2.129 enum RegionDestinationEnums

< Control the destination of the selected region.

##### Enumerator

***RegionDestination\_Stream0*** The destination of the region is the data stream 0.  
***RegionDestination\_Stream1*** The destination of the region is the data stream 1.  
***RegionDestination\_Stream2*** The destination of the region is the data stream 2.  
***NUM\_REGIONDESTINATION***

## 8.8.2.130 enum RegionModeEnums

< Controls if the selected Region of interest is active and streaming.

## Enumerator

**RegionMode\_Off** Disable the usage of the Region.

**RegionMode\_On** Enable the usage of the Region.

**NUM\_REGIONMODE**

## 8.8.2.131 enum RegionSelectorEnums

< Selects the Region of interest to control. The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently.

## Enumerator

**RegionSelector\_Region0** Selected feature will control the region 0.

**RegionSelector\_Region1** Selected feature will control the region 1.

**RegionSelector\_Region2** Selected feature will control the region 2.

**RegionSelector\_All** Selected features will control all the regions at the same time.

**NUM\_REGIONSELECTOR**

## 8.8.2.132 enum RgbTransformLightSourceEnums

< Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources. Selecting a value also sets the white balance ratios (BalanceRatioRed and BalanceRatioBlue), but those can be overwritten through manual or auto white balance.

## Enumerator

**RgbTransformLightSource\_General** Uses a matrix calibrated for a wide range of light sources.

**RgbTransformLightSource\_Tungsten2800K** Uses a matrix optimized for tungsten/incandescent light with color temperature 2800K.

**RgbTransformLightSource\_WarmFluorescent3000K** Uses a matrix optimized for a typical warm fluorescent light with color temperature 3000K.

**RgbTransformLightSource\_CoolFluorescent4000K** Uses a matrix optimized for a typical cool fluorescent light with color temperature 4000K.

**RgbTransformLightSource\_Daylight5000K** Uses a matrix optimized for noon Daylight with color temperature 5000K.

**RgbTransformLightSource\_Cloudy6500K** Uses a matrix optimized for a cloudy sky with color temperature 6500K.

**RgbTransformLightSource\_Shade8000K** Uses a matrix optimized for shade with color temperature 8000K.

**RgbTransformLightSource\_Custom** Uses a custom matrix set by the user through the Color↔TransformationValueSelector and ColorTransformationValue controls.

**NUM\_RGBTRANSFORMLIGHTSOURCE**

## 8.8.2.133 enum Scan3dCoordinateReferenceSelectorEnums

< Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.

## Enumerator

**Scan3dCoordinateReferenceSelector\_RotationX** Rotation around X axis.  
**Scan3dCoordinateReferenceSelector\_RotationY** Rotation around Y axis.  
**Scan3dCoordinateReferenceSelector\_RotationZ** Rotation around Z axis.  
**Scan3dCoordinateReferenceSelector\_TranslationX** X axis translation.  
**Scan3dCoordinateReferenceSelector\_TranslationY** Y axis translation.  
**Scan3dCoordinateReferenceSelector\_TranslationZ** Z axis translation.  
**NUM\_SCAN3DCOORDINATEREFERENCESELECTOR**

## 8.8.2.134 enum Scan3dCoordinateSelectorEnums

< Selects the individual coordinates in the vectors for 3D information/transformation.

## Enumerator

**Scan3dCoordinateSelector\_CoordinateA** The first (X or Theta) coordinate  
**Scan3dCoordinateSelector\_CoordinateB** The second (Y or Phi) coordinate  
**Scan3dCoordinateSelector\_CoordinateC** The third (Z or Rho) coordinate.  
**NUM\_SCAN3DCOORDINATESELECTOR**

## 8.8.2.135 enum Scan3dCoordinateSystemEnums

< Specifies the Coordinate system to use for the device.

## Enumerator

**Scan3dCoordinateSystem\_Cartesian** Default value. 3-axis orthogonal, right-hand X-Y-Z.  
**Scan3dCoordinateSystem\_Spherical** A Theta-Phi-Rho coordinate system.  
**Scan3dCoordinateSystem\_Cylindrical** A Theta-Y-Rho coordinate system.  
**NUM\_SCAN3DCOORDINATESYSTEM**

## 8.8.2.136 enum Scan3dCoordinateSystemReferenceEnums

< Defines coordinate system reference location.

## Enumerator

**Scan3dCoordinateSystemReference\_Anchor** Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.  
**Scan3dCoordinateSystemReference\_Transformed** Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices.  
**NUM\_SCAN3DCOORDINATESYSTEMREFERENCE**

## 8.8.2.137 enum Scan3dCoordinateTransformSelectorEnums

< Sets the index to read/write a coordinate transform value.

## Enumerator

**Scan3dCoordinateTransformSelector\_RotationX** Rotation around X axis.  
**Scan3dCoordinateTransformSelector\_RotationY** Rotation around Y axis.  
**Scan3dCoordinateTransformSelector\_RotationZ** Rotation around Z axis.  
**Scan3dCoordinateTransformSelector\_TranslationX** Translation along X axis.  
**Scan3dCoordinateTransformSelector\_TranslationY** Translation along Y axis.  
**Scan3dCoordinateTransformSelector\_TranslationZ** Translation along Z axis.  
**NUM\_SCAN3DCOORDINATETRANSFORMSELECTOR**

## 8.8.2.138 enum Scan3dDistanceUnitEnums

< Specifies the unit used when delivering calibrated distance data.

## Enumerator

**Scan3dDistanceUnit\_Millimeter** Distance values are in millimeter units (default).  
**Scan3dDistanceUnit\_Inch** Distance values are in inch units.  
**NUM\_SCAN3DDISTANCEUNIT**

## 8.8.2.139 enum Scan3dOutputModeEnums

< Controls the Calibration and data organization of the device, naming the coordinates transmitted.

## Enumerator

**Scan3dOutputMode\_UncalibratedC** Uncalibrated 2.5D Depth map. The distance data does not represent a physical unit and may be non-linear. The data is a 2.5D range map only.  
**Scan3dOutputMode\_CalibratedABC\_Grid** 3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.  
**Scan3dOutputMode\_CalibratedABC\_PointCloud** 3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.  
**Scan3dOutputMode\_CalibratedAC** 2 Coordinates with fixed B sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis uses the scale and offset parameters for the B axis.  
**Scan3dOutputMode\_CalibratedAC\_Linescan** 2 Coordinates with varying sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis comes from the encoder chunk value.  
**Scan3dOutputMode\_CalibratedC** Calibrated 2.5D Depth map. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. No information on X-Y axes available.  
**Scan3dOutputMode\_CalibratedC\_Linescan** Depth Map with varying B sampling. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. The B (Y) axis comes from the encoder chunk value.  
**Scan3dOutputMode\_RectifiedC** Rectified 2.5D Depth map. The distance data has been rectified to a uniform sampling pattern in the X and Y direction. The data is a 2.5D range map only. If a complete 3D point cloud is rectified but transmitted as explicit coordinates it should be transmitted as one of the "CalibratedABC" formats.

**Scan3dOutputMode\_RectifiedC\_Linescan** Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using Coord3D\_C pixels. The B (Y) axis comes from the encoder chunk value.

**Scan3dOutputMode\_DisparityC** Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.

**Scan3dOutputMode\_DisparityC\_Linescan** Disparity 2.5D Depth map with varying B sampling. The distance is inversely proportional to the pixel (disparity) value. The B (Y) axis comes from the encoder chunk value.

**NUM\_SCAN3DOUTPUTMODE**

#### 8.8.2.140 enum SensorDigitizationTapsEnums

< Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

Enumerator

**SensorDigitizationTaps\_One** 1 tap.

**SensorDigitizationTaps\_Two** 2 taps.

**SensorDigitizationTaps\_Three** 3 taps.

**SensorDigitizationTaps\_Four** 4 taps.

**SensorDigitizationTaps\_Eight** 8 taps.

**SensorDigitizationTaps\_Ten** 10 taps.

**NUM\_SENSORDIGITIZATIONTAPS**

#### 8.8.2.141 enum SensorShutterModeEnums

< Sets the shutter mode of the device.

Enumerator

**SensorShutterMode\_Global** The shutter opens and closes at the same time for all pixels. All the pixels are exposed for the same length of time at the same time.

**SensorShutterMode\_Rolling** The shutter opens and closes sequentially for groups (typically lines) of pixels. All the pixels are exposed for the same length of time but not at the same time.

**SensorShutterMode\_GlobalReset** The shutter opens at the same time for all pixels but ends in a sequential manner. The pixels are exposed for different lengths of time.

**NUM\_SENSORSHUTTERMODE**

#### 8.8.2.142 enum SensorTapsEnums

< Number of taps of the camera sensor.

Enumerator

**SensorTaps\_One** 1 tap.

**SensorTaps\_Two** 2 taps.

**SensorTaps\_Three** 3 taps.

**SensorTaps\_Four** 4 taps.

**SensorTaps\_Eight** 8 taps.

**SensorTaps\_Ten** 10 taps.

**NUM\_SENSORTAPS**

## 8.8.2.143 enum SequencerConfigurationModeEnums

< Controls whether or not a sequencer is in configuration mode.

Enumerator

***SequencerConfigurationMode\_Off***  
***SequencerConfigurationMode\_On***  
***NUM\_SEQUENCERCONFIGURATIONMODE***

## 8.8.2.144 enum SequencerConfigurationValidEnums

< Display whether the current sequencer configuration is valid to run.

Enumerator

***SequencerConfigurationValid\_No***  
***SequencerConfigurationValid\_Yes***  
***NUM\_SEQUENCERCONFIGURATIONVALID***

## 8.8.2.145 enum SequencerModeEnums

< Controls whether or not a sequencer is active.

Enumerator

***SequencerMode\_Off***  
***SequencerMode\_On***  
***NUM\_SEQUENCERMODE***

## 8.8.2.146 enum SequencerSetValidEnums

< Displays whether the currently selected sequencer set's register contents are valid to use.

Enumerator

***SequencerSetValid\_No***  
***SequencerSetValid\_Yes***  
***NUM\_SEQUENCERSETVALID***

## 8.8.2.147 enum SequencerTriggerActivationEnums

< Specifies the activation mode of the sequencer trigger.

Enumerator

***SequencerTriggerActivation\_RisingEdge***  
***SequencerTriggerActivation\_FallingEdge***  
***SequencerTriggerActivation\_AnyEdge***  
***SequencerTriggerActivation\_LevelHigh***  
***SequencerTriggerActivation\_LevelLow***  
***NUM\_SEQUENCERTRIGGERACTIVATION***



## 8.8.2.148 enum SequencerTriggerSourceEnums

< Specifies the internal signal or physical input line to use as the sequencer trigger source.

Enumerator

***SequencerTriggerSource\_Off***  
***SequencerTriggerSource\_FrameStart***  
***NUM\_SEQUENCERTRIGGERSOURCE***

## 8.8.2.149 enum SerialPortBaudRateEnums

< This feature controls the baud rate used by the selected serial port.

Enumerator

***SerialPortBaudRate\_Baud300***  
***SerialPortBaudRate\_Baud600***  
***SerialPortBaudRate\_Baud1200***  
***SerialPortBaudRate\_Baud2400***  
***SerialPortBaudRate\_Baud4800***  
***SerialPortBaudRate\_Baud9600***  
***SerialPortBaudRate\_Baud14400***  
***SerialPortBaudRate\_Baud19200***  
***SerialPortBaudRate\_Baud38400***  
***SerialPortBaudRate\_Baud57600***  
***SerialPortBaudRate\_Baud115200***  
***SerialPortBaudRate\_Baud230400***  
***SerialPortBaudRate\_Baud460800***  
***SerialPortBaudRate\_Baud921600***  
***NUM\_SERIALPORTBAUDRATE***

## 8.8.2.150 enum SerialPortParityEnums

< This feature controls the parity used by the selected serial port.

Enumerator

***SerialPortParity\_None***  
***SerialPortParity\_Odd***  
***SerialPortParity\_Even***  
***SerialPortParity\_Mark***  
***SerialPortParity\_Space***  
***NUM\_SERIALPORTPARITY***

#### 8.8.2.151 enum SerialPortSelectorEnums

< Selects which serial port of the device to control.

Enumerator

***SerialPortSelector\_SerialPort0***  
***NUM\_SERIALPORTSELECTOR***

#### 8.8.2.152 enum SerialPortSourceEnums

< Specifies the physical input Line on which to receive serial data.

Enumerator

***SerialPortSource\_Line0***  
***SerialPortSource\_Line1***  
***SerialPortSource\_Line2***  
***SerialPortSource\_Line3***  
***SerialPortSource\_Off***  
***NUM\_SERIALPORTSOURCE***

#### 8.8.2.153 enum SerialPortStopBitsEnums

< This feature controls the number of stop bits used by the selected serial port.

Enumerator

***SerialPortStopBits\_Bits1***  
***SerialPortStopBits\_Bits1AndAHalf***  
***SerialPortStopBits\_Bits2***  
***NUM\_SERIALPORTSTOPBITS***

#### 8.8.2.154 enum SoftwareSignalSelectorEnums

< Selects which Software Signal features to control.

Enumerator

***SoftwareSignalSelector\_SoftwareSignal0*** Selects the software generated signal to control.  
***SoftwareSignalSelector\_SoftwareSignal1*** Selects the software generated signal to control.  
***SoftwareSignalSelector\_SoftwareSignal2*** Selects the software generated signal to control.  
***NUM\_SOFTWARESIGNALSELECTOR***

## 8.8.2.155 enum SourceSelectorEnums

< Selects the source to control.

## Enumerator

**SourceSelector\_Source0** Selects the data source 0.

**SourceSelector\_Source1** Selects the data source 1.

**SourceSelector\_Source2** Selects the data source 2.

**SourceSelector\_All** Selects all the data sources.

**NUM\_SOURCESELECTOR**

## 8.8.2.156 enum TestPatternEnums

< Selects the type of test pattern that is generated by the device as image source.

## Enumerator

**TestPattern\_Off** Test pattern is disabled.

**TestPattern\_Increment** Pixel value increments by 1 for each pixel.

**TestPattern\_SensorTestPattern** A test pattern generated by the image sensor. The pattern varies for different sensor models.

**NUM\_TESTPATTERN**

## 8.8.2.157 enum TestPatternGeneratorSelectorEnums

< Selects which test pattern generator is controlled by the TestPattern feature.

## Enumerator

**TestPatternGeneratorSelector\_Sensor** TestPattern feature controls the sensor's test pattern generator.

**TestPatternGeneratorSelector\_PipelineStart** TestPattern feature controls the test pattern inserted at the start of the image pipeline.

**NUM\_TESTPATTERNGENERATORSELECTOR**

## 8.8.2.158 enum TimerSelectorEnums

< Selects which Timer to configure.

## Enumerator

**TimerSelector\_Timer0** Selects the Timer 0.

**TimerSelector\_Timer1** Selects the Timer 1.

**TimerSelector\_Timer2** Selects the Timer 2.

**NUM\_TIMERSELECTOR**

## 8.8.2.159 enum TimerStatusEnums

< Returns the current status of the Timer.

## Enumerator

**TimerStatus\_TimerIdle** The Timer is idle.  
**TimerStatus\_TimerTriggerWait** The Timer is waiting for a start trigger.  
**TimerStatus\_TimerActive** The Timer is counting for the specified duration.  
**TimerStatus\_TimerCompleted** The Timer reached the TimerDuration count.  
**NUM\_TIMERSTATUS**

## 8.8.2.160 enum TimerTriggerActivationEnums

< Selects the activation mode of the trigger to start the Timer.

## Enumerator

**TimerTriggerActivation\_RisingEdge** Starts counting on the Rising Edge of the selected trigger signal.  
**TimerTriggerActivation\_FallingEdge** Starts counting on the Falling Edge of the selected trigger signal.  
**TimerTriggerActivation\_AnyEdge** Starts counting on the Falling or Rising Edge of the selected trigger signal.  
**TimerTriggerActivation\_LevelHigh** Counts as long as the selected trigger signal level is High.  
**TimerTriggerActivation\_LevelLow** Counts as long as the selected trigger signal level is Low.  
**NUM\_TIMERTRIGGERACTIVATION**

## 8.8.2.161 enum TimerTriggerSourceEnums

< Selects the source of the trigger to start the Timer.

## Enumerator

**TimerTriggerSource\_Off** Disables the Timer trigger.  
**TimerTriggerSource\_AcquisitionTrigger** Starts with the reception of the Acquisition Trigger.  
**TimerTriggerSource\_AcquisitionStart** Starts with the reception of the Acquisition Start.  
**TimerTriggerSource\_AcquisitionEnd** Starts with the reception of the Acquisition End.  
**TimerTriggerSource\_FrameTrigger** Starts with the reception of the Frame Start Trigger.  
**TimerTriggerSource\_FrameStart** Starts with the reception of the Frame Start.  
**TimerTriggerSource\_FrameEnd** Starts with the reception of the Frame End.  
**TimerTriggerSource\_FrameBurstStart** Starts with the reception of the Frame Burst Start.  
**TimerTriggerSource\_FrameBurstEnd** Starts with the reception of the Frame Burst End.  
**TimerTriggerSource\_LineTrigger** Starts with the reception of the Line Start Trigger.  
**TimerTriggerSource\_LineStart** Starts with the reception of the Line Start.  
**TimerTriggerSource\_LineEnd** Starts with the reception of the Line End.  
**TimerTriggerSource\_ExposureStart** Starts with the reception of the Exposure Start.  
**TimerTriggerSource\_ExposureEnd** Starts with the reception of the Exposure End.

**TimerTriggerSource\_Line0** Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.

**TimerTriggerSource\_Line1** Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.

**TimerTriggerSource\_Line2** Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.

**TimerTriggerSource\_UserOutput0** Specifies which User Output bit signal to use as internal source for the trigger.

**TimerTriggerSource\_UserOutput1** Specifies which User Output bit signal to use as internal source for the trigger.

**TimerTriggerSource\_UserOutput2** Specifies which User Output bit signal to use as internal source for the trigger.

**TimerTriggerSource\_Counter0Start** Starts with the reception of the Counter Start.

**TimerTriggerSource\_Counter1Start** Starts with the reception of the Counter Start.

**TimerTriggerSource\_Counter2Start** Starts with the reception of the Counter Start.

**TimerTriggerSource\_Counter0End** Starts with the reception of the Counter End.

**TimerTriggerSource\_Counter1End** Starts with the reception of the Counter End.

**TimerTriggerSource\_Counter2End** Starts with the reception of the Counter End.

**TimerTriggerSource\_Timer0Start** Starts with the reception of the Timer Start.

**TimerTriggerSource\_Timer1Start** Starts with the reception of the Timer Start.

**TimerTriggerSource\_Timer2Start** Starts with the reception of the Timer Start.

**TimerTriggerSource\_Timer0End** Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.

**TimerTriggerSource\_Timer1End** Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.

**TimerTriggerSource\_Timer2End** Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.

**TimerTriggerSource\_Encoder0** Starts with the reception of the Encoder output signal.

**TimerTriggerSource\_Encoder1** Starts with the reception of the Encoder output signal.

**TimerTriggerSource\_Encoder2** Starts with the reception of the Encoder output signal.

**TimerTriggerSource\_SoftwareSignal0** Starts on the reception of the Software Signal.

**TimerTriggerSource\_SoftwareSignal1** Starts on the reception of the Software Signal.

**TimerTriggerSource\_SoftwareSignal2** Starts on the reception of the Software Signal.

**TimerTriggerSource\_Action0** Starts with the assertion of the chosen action signal.

**TimerTriggerSource\_Action1** Starts with the assertion of the chosen action signal.

**TimerTriggerSource\_Action2** Starts with the assertion of the chosen action signal.

**TimerTriggerSource\_LinkTrigger0** Starts with the reception of the chosen Link Trigger.

**TimerTriggerSource\_LinkTrigger1** Starts with the reception of the chosen Link Trigger.

**TimerTriggerSource\_LinkTrigger2** Starts with the reception of the chosen Link Trigger.

**NUM\_TIMERTRIGGERSOURCE**

#### 8.8.2.162 enum TransferComponentSelectorEnums

< Selects the color component for the control of the TransferStreamChannel feature.

## Enumerator

***TransferComponentSelector\_Red*** The TransferStreamChannel feature controls the index of the stream channel for the streaming of the red plane of the planar pixel formats.

***TransferComponentSelector\_Green*** The TransferStreamChannel feature controls the index of the stream channel for the streaming of the green plane of the planar pixel formats.

***TransferComponentSelector\_Blue*** The TransferStreamChannel feature controls the index of the stream channel for the streaming of blue plane of the planar pixel formats.

***TransferComponentSelector\_All*** The TransferStreamChannel feature controls the index of the stream channel for the streaming of all the planes of the planar pixel formats simultaneously or non planar pixel formats.

***NUM\_TRANSFERCOMPONENTSELECTOR***

## 8.8.2.163 enum TransferControlModeEnums

< Selects the control method for the transfers. Basic and Automatic start transmitting data as soon as there is enough data to fill a link layer packet. User Controlled allows you to directly control the transfer of blocks.

## Enumerator

***TransferControlMode\_Basic*** Basic

***TransferControlMode\_Automatic*** Automatic

***TransferControlMode\_UserControlled*** User Controlled

***NUM\_TRANSFERCONTROLMODE***

## 8.8.2.164 enum TransferOperationModeEnums

< Selects the operation mode of the transfer. Continuous is similar to Basic/Automatic but you can start/stop the transfer while acquisition runs independently. Multi Block transmits a specified number of blocks and then stops.

## Enumerator

***TransferOperationMode\_Continuous*** Continuous

***TransferOperationMode\_MultiBlock*** Multi Block

***NUM\_TRANSFEROPERATIONMODE***

## 8.8.2.165 enum TransferQueueModeEnums

< Specifies the operation mode of the transfer queue.

## Enumerator

***TransferQueueMode\_FirstInFirstOut*** Blocks first In are transferred Out first.

***NUM\_TRANSFERQUEUEMODE***

## 8.8.2.166 enum TransferSelectorEnums

< Selects which stream transfers are currently controlled by the selected Transfer features.

## Enumerator

***TransferSelector\_Stream0*** The transfer features control the data stream 0.  
***TransferSelector\_Stream1*** The transfer features control the data stream 1.  
***TransferSelector\_Stream2*** The transfer features control the data stream 2.  
***TransferSelector\_All*** The transfer features control all the data streams simulateneously.  
***NUM\_TRANSFERSELECTOR***

## 8.8.2.167 enum TransferStatusSelectorEnums

< Selects which status of the transfer module to read.

## Enumerator

***TransferStatusSelector\_Streaming*** Data blocks are transmitted when enough data is available.  
***TransferStatusSelector\_Paused*** Data blocks transmission is suspended immediately.  
***TransferStatusSelector\_Stopping*** Data blocks transmission is stopping. The current block transmission will be completed and the transfer state will stop.  
***TransferStatusSelector\_Stopped*** Data blocks transmission is stopped.  
***TransferStatusSelector\_QueueOverflow*** Data blocks queue is in overflow state.  
***NUM\_TRANSFERSTATUSSELECTOR***

## 8.8.2.168 enum TransferTriggerActivationEnums

< Specifies the activation mode of the transfer control trigger.

## Enumerator

***TransferTriggerActivation\_RisingEdge*** Specifies that the trigger is considered valid on the rising edge of the source signal.  
***TransferTriggerActivation\_FallingEdge*** Specifies that the trigger is considered valid on the falling edge of the source signal.  
***TransferTriggerActivation\_AnyEdge*** Specifies that the trigger is considered valid on the falling or rising edge of the source signal.  
***TransferTriggerActivation\_LevelHigh*** Specifies that the trigger is considered valid as long as the level of the source signal is high. This can apply to TransferActive and TransferPause trigger.  
***TransferTriggerActivation\_LevelLow*** Specifies that the trigger is considered valid as long as the level of the source signal is low. This can apply to TransferActive and TransferPause trigger.  
***NUM\_TRANSFERTRIGGERACTIVATION***

## 8.8.2.169 enum TransferTriggerModeEnums

< Controls if the selected trigger is active.

## Enumerator

**TransferTriggerMode\_Off** Disables the selected trigger.

**TransferTriggerMode\_On** Enable the selected trigger.

**NUM\_TRANSFERTRIGGERMODE**

## 8.8.2.170 enum TransferTriggerSelectorEnums

< Selects the type of transfer trigger to configure.

## Enumerator

**TransferTriggerSelector\_TransferStart** Selects a trigger to start the transfers.

**TransferTriggerSelector\_TransferStop** Selects a trigger to stop the transfers.

**TransferTriggerSelector\_TransferAbort** Selects a trigger to abort the transfers.

**TransferTriggerSelector\_TransferPause** Selects a trigger to pause the transfers.

**TransferTriggerSelector\_TransferResume** Selects a trigger to Resume the transfers.

**TransferTriggerSelector\_TransferActive** Selects a trigger to Activate the transfers. This trigger type is used when TriggerActivation is set LevelHigh or levelLow.

**TransferTriggerSelector\_TransferBurstStart** Selects a trigger to start the transfer of a burst of frames specified by TransferBurstCount.

**TransferTriggerSelector\_TransferBurstStop** Selects a trigger to end the transfer of a burst of frames.

**NUM\_TRANSFERTRIGGERSELECTOR**

## 8.8.2.171 enum TransferTriggerSourceEnums

< Specifies the signal to use as the trigger source for transfers.

## Enumerator

**TransferTriggerSource\_Line0** Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.

**TransferTriggerSource\_Line1** Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.

**TransferTriggerSource\_Line2** Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.

**TransferTriggerSource\_Counter0Start** Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.

**TransferTriggerSource\_Counter1Start** Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.

**TransferTriggerSource\_Counter2Start** Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.

**TransferTriggerSource\_Counter0End** Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.



- TransferTriggerSource\_Counter1End*** Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Counter2End*** Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Timer0Start*** Specifies which Timer signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Timer1Start*** Specifies which Timer signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Timer2Start*** Specifies which Timer signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Timer0End*** Specifies which Timer signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Timer1End*** Specifies which Timer signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Timer2End*** Specifies which Timer signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_SoftwareSignal0*** Specifies which Software Signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_SoftwareSignal1*** Specifies which Software Signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_SoftwareSignal2*** Specifies which Software Signal to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Action0*** Specifies which Action command to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Action1*** Specifies which Action command to use as internal source for the transfer control trigger signal.
- TransferTriggerSource\_Action2*** Specifies which Action command to use as internal source for the transfer control trigger signal.
- NUM\_TRANSFERTRIGGERSOURCE***

#### 8.8.2.172 enum TriggerActivationEnums

< Specifies the activation mode of the trigger.

Enumerator

***TriggerActivation\_LevelLow***  
***TriggerActivation\_LevelHigh***  
***TriggerActivation\_FallingEdge***  
***TriggerActivation\_RisingEdge***  
***TriggerActivation\_AnyEdge***  
***NUM\_TRIGGERACTIVATION***

#### 8.8.2.173 enum TriggerModeEnums

< Controls whether or not trigger is active.

Enumerator

***TriggerMode\_Off***  
***TriggerMode\_On***  
***NUM\_TRIGGERMODE***

## 8.8.2.174 enum TriggerOverlapEnums

< Specifies the overlap mode of the trigger.

Enumerator

***TriggerOverlap\_Off***  
***TriggerOverlap\_ReadOut***  
***TriggerOverlap\_PreviousFrame***  
***NUM\_TRIGGEROVERLAP***

## 8.8.2.175 enum TriggerSelectorEnums

< Selects the type of trigger to configure.

Enumerator

***TriggerSelector\_AcquisitionStart***  
***TriggerSelector\_FrameStart***  
***TriggerSelector\_FrameBurstStart***  
***NUM\_TRIGGERSELECTOR***

## 8.8.2.176 enum TriggerSourceEnums

< Specifies the internal signal or physical input line to use as the trigger source.

Enumerator

***TriggerSource\_Software***  
***TriggerSource\_Line0***  
***TriggerSource\_Line1***  
***TriggerSource\_Line2***  
***TriggerSource\_Line3***  
***TriggerSource\_UserOutput0***  
***TriggerSource\_UserOutput1***  
***TriggerSource\_UserOutput2***  
***TriggerSource\_UserOutput3***  
***TriggerSource\_Counter0Start***  
***TriggerSource\_Counter1Start***  
***TriggerSource\_Counter0End***  
***TriggerSource\_Counter1End***  
***TriggerSource\_LogicBlock0***  
***TriggerSource\_LogicBlock1***  
***TriggerSource\_Action0***  
***NUM\_TRIGGERSOURCE***

## 8.8.2.177 enum UserOutputSelectorEnums

< Selects which bit of the User Output register is set by UserOutputValue.

Enumerator

***UserOutputSelector\_UserOutput0***  
***UserOutputSelector\_UserOutput1***  
***UserOutputSelector\_UserOutput2***  
***UserOutputSelector\_UserOutput3***  
***NUM\_USEROUTPUTSELECTOR***

## 8.8.2.178 enum UserSetDefaultEnums

< Selects the feature User Set to load and make active by default when the device is restarted.

Enumerator

***UserSetDefault\_Default*** Factory default set.  
***UserSetDefault\_UserSet0*** User configurable set 0.  
***UserSetDefault\_UserSet1*** User configurable set 1.  
***NUM\_USERSETDEFAULT***

## 8.8.2.179 enum UserSetSelectorEnums

< Selects the feature User Set to load, save or configure.

Enumerator

***UserSetSelector\_Default*** Factory default set.  
***UserSetSelector\_UserSet0*** User configurable set 0.  
***UserSetSelector\_UserSet1*** User configurable set 1.  
***NUM\_USERSETSELECTOR***

## 8.8.2.180 enum WhiteClipSelectorEnums

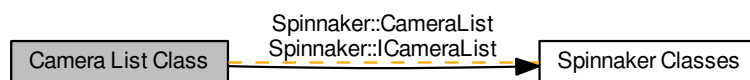
< Selects which White Clip to control.

Enumerator

***WhiteClipSelector\_All*** White Clip will be applied to all channels or taps.  
***WhiteClipSelector\_Red*** White Clip will be applied to the red channel.  
***WhiteClipSelector\_Green*** White Clip will be applied to the green channel.  
***WhiteClipSelector\_Blue*** White Clip will be applied to the blue channel.  
***WhiteClipSelector\_Y*** White Clip will be applied to Y channel.  
***WhiteClipSelector\_U*** White Clip will be applied to U channel.  
***WhiteClipSelector\_V*** White Clip will be applied to V channel.  
***WhiteClipSelector\_Tap1*** White Clip will be applied to Tap 1.  
***WhiteClipSelector\_Tap2*** White Clip will be applied to Tap 2.  
***NUM\_WHITECLIPSELECTOR***

## 8.9 Camera List Class

Collaboration diagram for Camera List Class:



### Classes

- class [CameraList](#)  
*Used to hold a list of camera objects.*
- class [ICameraList](#)  
*Used to hold a list of camera objects.*

### 8.9.1 Detailed Description

## 8.10 CameraPtr Class

Collaboration diagram for CameraPtr Class:



### Classes

- class [CameraPtr](#)  
*A reference tracked pointer to a camera object.*

### Functions

- [CameraPtr](#) () throw ()  
*Default constructor.*
- [CameraPtr](#) (const int) throw ()  
*Default constructor.*
- virtual [~CameraPtr](#) (void)  
*Virtual destructor.*
- virtual CameraPtr & [operator=](#) (const int nMustBeNull)  
*Assignment operator.*

#### 8.10.1 Detailed Description

#### 8.10.2 Function Documentation

##### 8.10.2.1 CameraPtr ( ) throw ) [inline]

Default constructor.

##### 8.10.2.2 CameraPtr ( const int ) throw ) [inline]

Default constructor.

##### 8.10.2.3 virtual CameraPtr& operator= ( const int nMustBeNull ) [inline],[virtual]

Assignment operator.

Reimplemented from [BasePtr< Camera, ICameraBase >](#).

##### 8.10.2.4 virtual ~CameraPtr ( void ) [inline],[virtual]

Virtual destructor.

## 8.11 ChunkData Class

Collaboration diagram for ChunkData Class:



### Classes

- class [ChunkData](#)

*The chunk data which contains additional information about an image.*

#### 8.11.1 Detailed Description

## 8.12 DeviceEvent Class

Collaboration diagram for DeviceEvent Class:



### Classes

- class [DeviceEvent](#)  
*A handler to device events.*

### 8.12.1 Detailed Description

## 8.13 Event Class

Collaboration diagram for Event Class:



### Classes

- class [Event](#)

*The base class for all event types.*

#### 8.13.1 Detailed Description



## 8.14 Exception Class

Collaboration diagram for Exception Class:



### Classes

- class [Exception](#)

The [Exception](#) object represents an error that is returned from the library.

### 8.14.1 Detailed Description

## 8.15 Image Class

Collaboration diagram for Image Class:



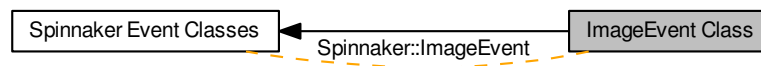
### Classes

- class [Image](#)  
*The image object class.*

#### 8.15.1 Detailed Description

## 8.16 ImageEvent Class

Collaboration diagram for ImageEvent Class:



### Classes

- class [ImageEvent](#)  
*A handler for capturing image arrival events.*

#### 8.16.1 Detailed Description

## 8.17 ImagePtr Class

Collaboration diagram for ImagePtr Class:



### Classes

- class [ImagePtr](#)  
*A reference tracked pointer to an image object.*

### 8.17.1 Detailed Description

## 8.18 ImageStatistics Class

Collaboration diagram for ImageStatistics Class:



### Classes

- class [ImageStatistics](#)  
*Represents image statistics for an image.*

#### 8.18.1 Detailed Description

## 8.19 Interface Class

Collaboration diagram for Interface Class:



### Classes

- class [Interface](#)

*An interface object which holds a list of cameras.*

### 8.19.1 Detailed Description

## 8.20 InterfaceEvent Class

Collaboration diagram for InterfaceEvent Class:



### Classes

- class [InterfaceEvent](#)  
*A handler to device arrival and removal events on all interfaces.*

### 8.20.1 Detailed Description

## 8.21 InterfaceList Class

Collaboration diagram for InterfaceList Class:



### Classes

- class [InterfaceList](#)  
*A list of the available interfaces on the system.*

### 8.21.1 Detailed Description



## 8.22 InterfacePtr Class

Collaboration diagram for InterfacePtr Class:



### Classes

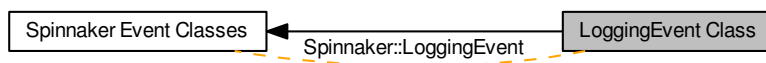
- class [InterfacePtr](#)

*A reference tracked pointer to the interface object.*

### 8.22.1 Detailed Description

## 8.23 LoggingEvent Class

Collaboration diagram for LoggingEvent Class:



### Classes

- class [LoggingEvent](#)  
*An event handler for capturing the device logging event.*

#### 8.23.1 Detailed Description

## 8.24 Logging Event Class

Collaboration diagram for Logging Event Class:



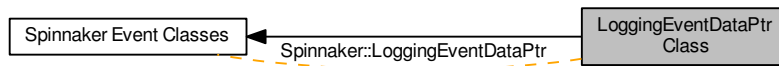
### Classes

- class [LoggingEventData](#)  
The [LoggingEventData](#) object.

#### 8.24.1 Detailed Description

## 8.25 LoggingEventDataPtr Class

Collaboration diagram for LoggingEventDataPtr Class:



### Classes

- class [LoggingEventDataPtr](#)  
*A reference tracked pointer to the [LoggingEvent](#) object.*

### 8.25.1 Detailed Description

## 8.26 RemovalEvent Class

Collaboration diagram for RemovalEvent Class:



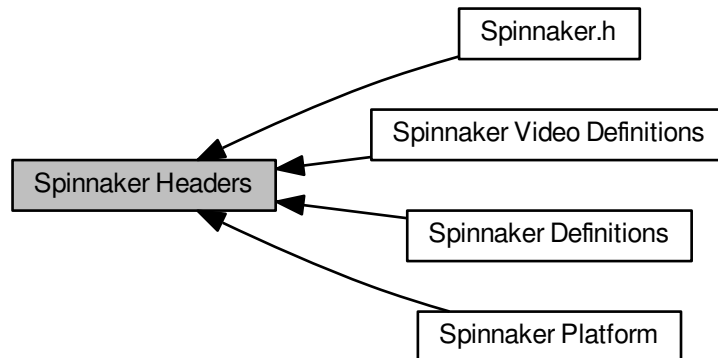
### Classes

- class [RemovalEvent](#)  
*An event handler for capturing the device removal event.*

#### 8.26.1 Detailed Description

## 8.27 Spinnaker Headers

Collaboration diagram for Spinnaker Headers:



### Modules

- [Spinnaker.h](#)  
*Global header file for [Spinnaker](#).*
- [Spinnaker Definitions](#)  
*Definitions file for [Spinnaker](#).*
- [Spinnaker Platform](#)  
*Platform-specific header file for [Spinnaker](#).*
- [Spinnaker Video Definitions](#)  
*Definitions file for [Spinnaker](#) video recorder.*

### Classes

- struct [MJPGOption](#)  
*Options for saving MJPG files.*
- struct [H264Option](#)  
*Options for saving H264 files.*
- struct [AVIOption](#)  
*Options for saving AVI files.*

### Variables

- const uint64\_t [EVENT\\_TIMEOUT\\_NONE](#) = 0  
*Timeout values for getting next image, device, or interface event.*
- const uint64\_t [EVENT\\_TIMEOUT\\_INFINITE](#) = 0xFFFFFFFFFFFFFFFF

### 8.27.1 Detailed Description

### 8.27.2 Variable Documentation

8.27.2.1 `const uint64_t EVENT_TIMEOUT_INFINITE = 0xFFFFFFFFFFFFFFFF`

8.27.2.2 `const uint64_t EVENT_TIMEOUT_NONE = 0`

Timeout values for getting next image, device, or interface event.

## 8.28 Spinnaker.h

Global header file for [Spinnaker](#).

Collaboration diagram for Spinnaker.h:



Global header file for [Spinnaker](#).

By including this file, all required header files for full [Spinnaker](#) operation will be included automatically. It is recommended that this file be used instead of manually including individual header files.

We welcome your bug reports, suggestions, and comments: <http://www.ptgrey.com/support/contact-us>



## 8.29 Spinnaker Definitions

Definitions file for [Spinnaker](#).

Collaboration diagram for Spinnaker Definitions:



### Classes

- struct [PNGOption](#)  
*Options for saving PNG images.*
- struct [PPMOption](#)  
*Options for saving PPM images.*
- struct [PGMOption](#)  
*Options for saving PGM images.*
- struct [TIFFOption](#)  
*Options for saving TIFF images.*
- struct [JPEGOption](#)  
*Options for saving JPEG image.*
- struct [JPG2Option](#)  
*Options for saving JPEG2000 image.*
- struct [BMPOption](#)  
*Options for saving Bitmap image.*
- struct [LibraryVersion](#)  
*Provides easier access to the current version of [Spinnaker](#).*
- struct [ActionCommandResult](#)  
*Action Command Result.*

## Enumerations

- enum `Error` {
  - `SPINNAKER_ERR_SUCCESS` = 0,
  - `SPINNAKER_ERR_ERROR` = -1001,
  - `SPINNAKER_ERR_NOT_INITIALIZED` = -1002,
  - `SPINNAKER_ERR_NOT_IMPLEMENTED` = -1003,
  - `SPINNAKER_ERR_RESOURCE_IN_USE` = -1004,
  - `SPINNAKER_ERR_ACCESS_DENIED` = -1005,
  - `SPINNAKER_ERR_INVALID_HANDLE` = -1006,
  - `SPINNAKER_ERR_INVALID_ID` = -1007,
  - `SPINNAKER_ERR_NO_DATA` = -1008,
  - `SPINNAKER_ERR_INVALID_PARAMETER` = -1009,
  - `SPINNAKER_ERR_IO` = -1010,
  - `SPINNAKER_ERR_TIMEOUT` = -1011,
  - `SPINNAKER_ERR_ABORT` = -1012,
  - `SPINNAKER_ERR_INVALID_BUFFER` = -1013,
  - `SPINNAKER_ERR_NOT_AVAILABLE` = -1014,
  - `SPINNAKER_ERR_INVALID_ADDRESS` = -1015,
  - `SPINNAKER_ERR_BUFFER_TOO_SMALL` = -1016,
  - `SPINNAKER_ERR_INVALID_INDEX` = -1017,
  - `SPINNAKER_ERR_PARSING_CHUNK_DATA` = -1018,
  - `SPINNAKER_ERR_INVALID_VALUE` = -1019,
  - `SPINNAKER_ERR_RESOURCE_EXHAUSTED` = -1020,
  - `SPINNAKER_ERR_OUT_OF_MEMORY` = -1021,
  - `SPINNAKER_ERR_BUSY` = -1022,
  - `GENICAM_ERR_INVALID_ARGUMENT` = -2001,
  - `GENICAM_ERR_OUT_OF_RANGE` = -2002,
  - `GENICAM_ERR_PROPERTY` = -2003,
  - `GENICAM_ERR_RUN_TIME` = -2004,
  - `GENICAM_ERR_LOGICAL` = -2005,
  - `GENICAM_ERR_ACCESS` = -2006,
  - `GENICAM_ERR_TIMEOUT` = -2007,
  - `GENICAM_ERR_DYNAMIC_CAST` = -2008,
  - `GENICAM_ERR_GENERIC` = -2009,
  - `GENICAM_ERR_BAD_ALLOCATION` = -2010,
  - `SPINNAKER_ERR_IM_CONVERT` = -3001,
  - `SPINNAKER_ERR_IM_COPY` = -3002,
  - `SPINNAKER_ERR_IM_MALLOC` = -3003,
  - `SPINNAKER_ERR_IM_NOT_SUPPORTED` = -3004,
  - `SPINNAKER_ERR_IM_HISTOGRAM_RANGE` = -3005,
  - `SPINNAKER_ERR_IM_HISTOGRAM_MEAN` = -3006,
  - `SPINNAKER_ERR_IM_MIN_MAX` = -3007,
  - `SPINNAKER_ERR_IM_COLOR_CONVERSION` = -3008,
  - `SPINNAKER_ERR_IM_DECOMPRESSION` = -3009,
  - `SPINNAKER_ERR_CUSTOM_ID` = -10000 }

*Spinnaker enum definitions.*

- enum `EventType` {
  - `SPINNAKER_EVENT_ARRIVAL_REMOVAL`,
  - `SPINNAKER_EVENT_DEVICE`,
  - `SPINNAKER_EVENT_DEVICE_SPECIFIC`,
  - `SPINNAKER_EVENT_NEW_BUFFER`,
  - `SPINNAKER_EVENT_LOGGING_EVENT`,
  - `SPINNAKER_EVENT_UNKNOWN` }

*Event types in Spinnaker.*

- enum `PixelFormatNamespaceID` {

```

SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,
SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,
SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,
SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }

```

*This enum represents the namespace in which the TL specific pixel format resides.*

- enum `ColorProcessingAlgorithm` {  
`DEFAULT`,  
`NO_COLOR_PROCESSING`,  
`NEAREST_NEIGHBOR`,  
`EDGE_SENSING`,  
`HQ_LINEAR`,  
`RIGOROUS`,  
`IPP`,  
`DIRECTIONAL_FILTER`,  
`WEIGHTED_DIRECTIONAL_FILTER` }

*Color processing algorithms.*

- enum `PolarizationAlgorithm` {  
`NO_POLARIZATION`,  
`QUADRANT_I0_GRAYSCALE`,  
`QUADRANT_I45_GRAYSCALE`,  
`QUADRANT_I90_GRAYSCALE`,  
`QUADRANT_I135_GRAYSCALE`,  
`STOKES_S0_GRAYSCALE`,  
`STOKES_S0_HEATMAP`,  
`STOKES_S1_GRAYSCALE`,  
`STOKES_S1_HEATMAP`,  
`STOKES_S2_GRAYSCALE`,  
`STOKES_S2_HEATMAP`,  
`DOLP_GRAYSCALE`,  
`DOLP_HEATMAP`,  
`AOP_GRAYSCALE`,  
`AOP_HEATMAP` }
- enum `PolarizationResolution` {  
`QUARTER_RESOLUTION`,  
`FULL_RESOLUTION` }
- enum `HeatMapColor` {  
`HEATMAP_BLACK` = 1,  
`HEATMAP_BLUE` = 2,  
`HEATMAP_CYAN` = 3,  
`HEATMAP_GREEN` = 4,  
`HEATMAP_YELLOW` = 5,  
`HEATMAP_RED` = 6,  
`HEATMAP_WHITE` = 7 }
- enum `ImageFileFormat` {  
`FROM_FILE_EXT` = -1,  
`PGM`,  
`PPM`,  
`BMP`,  
`JPEG`,  
`JPEG2000`,  
`TIFF`,  
`PNG`,  
`RAW`,  
`JPEG12_C`,  
`IMAGE_FILE_FORMAT_FORCE_32BITS` = 0x7FFFFFFF }

*File formats to be used for saving images to disk.*

- enum `ImageStatus` {  
`IMAGE_UNKNOWN_ERROR` = -1,  
`IMAGE_NO_ERROR` = 0,  
`IMAGE_CRC_CHECK_FAILED` = 1,  
`IMAGE_DATA_OVERFLOW` = 2,  
`IMAGE_MISSING_PACKETS` = 3,  
`IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT` = 4,  
`IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT` = 5,  
`IMAGE_PACKETID_INCONSISTENT` = 6,  
`IMAGE_MISSING_LEADER` = 7,  
`IMAGE_MISSING_TRAILER` = 8,  
`IMAGE_DATA_INCOMPLETE` = 9,  
`IMAGE_INFO_INCONSISTENT` = 10,  
`IMAGE_CHUNK_DATA_INVALID` = 11,  
`IMAGE_NO_SYSTEM_RESOURCES` = 12 }

*Status of images returned from `GetNextImage()` call.*

- enum `StatisticsChannel` {  
`GREY`,  
`RED`,  
`GREEN`,  
`BLUE`,  
`HUE`,  
`SATURATION`,  
`LIGHTNESS`,  
`NUM_STATISTICS_CHANNELS` }

*Channels that allow statistics to be calculated.*

- enum `SpinnakerLogLevel` {  
`LOG_LEVEL_OFF` = -1,  
`LOG_LEVEL_FATAL` = 0,  
`LOG_LEVEL_ALERT` = 100,  
`LOG_LEVEL_CRIT` = 200,  
`LOG_LEVEL_ERROR` = 300,  
`LOG_LEVEL_WARN` = 400,  
`LOG_LEVEL_NOTICE` = 500,  
`LOG_LEVEL_INFO` = 600,  
`LOG_LEVEL_DEBUG` = 700,  
`LOG_LEVEL_NOTSET` = 800 }

*log levels*

- enum `PayloadTypeInfoIDs` {  
`PAYLOAD_TYPE_UNKNOWN` = 0,  
`PAYLOAD_TYPE_IMAGE` = 1,  
`PAYLOAD_TYPE_RAW_DATA` = 2,  
`PAYLOAD_TYPE_FILE` = 3,  
`PAYLOAD_TYPE_CHUNK_DATA` = 4,  
`PAYLOAD_TYPE_JPEG` = 5,  
`PAYLOAD_TYPE_JPEG2000` = 6,  
`PAYLOAD_TYPE_H264` = 7,  
`PAYLOAD_TYPE_CHUNK_ONLY` = 8,  
`PAYLOAD_TYPE_DEVICE_SPECIFIC` = 9,  
`PAYLOAD_TYPE_MULTI_PART` = 10,  
`PAYLOAD_TYPE_CUSTOM_ID` = 1000,  
`PAYLOAD_TYPE_EXTENDED_CHUNK` = 1001 }
- enum `ActionCommandStatus` {  
`ACTION_COMMAND_STATUS_OK` = 0,  
`ACTION_COMMAND_STATUS_NO_REF_TIME` = 0x8013,  
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,  
`ACTION_COMMAND_STATUS_ACTION_LATE` = 0x8016,  
`ACTION_COMMAND_STATUS_ERROR` = 0x8FFF }

*Possible Status Codes Returned from Action Command.*

- enum `PixelFormatIntType` {  
`IntType_UINT8`,  
`IntType_INT8`,  
`IntType_UINT10`,  
`IntType_UINT10p`,  
`IntType_UINT10P`,  
`IntType_UINT12`,  
`IntType_UINT12p`,  
`IntType_UINT12P`,  
`IntType_UINT14`,  
`IntType_UINT16`,  
`IntType_FLOAT32`,  
`IntType_UNKNOWN` }

*Possible integer types and packing used in a pixel format.*

### 8.29.1 Detailed Description

Definitions file for [Spinnaker](#).

### 8.29.2 Enumeration Type Documentation

#### 8.29.2.1 enum ActionCommandStatus

Possible Status Codes Returned from Action Command.

Enumerator

***ACTION\_COMMAND\_STATUS\_OK***  
***ACTION\_COMMAND\_STATUS\_NO\_REF\_TIME***  
***ACTION\_COMMAND\_STATUS\_OVERFLOW***  
***ACTION\_COMMAND\_STATUS\_ACTION\_LATE***  
***ACTION\_COMMAND\_STATUS\_ERROR***

#### 8.29.2.2 enum ColorProcessingAlgorithm

Color processing algorithms.

Please refer to our knowledge base at article at <http://www.ptgrey.com/support/kb/index.asp?a=4&q=33> for complete details for each algorithm.

Enumerator

***DEFAULT*** Default method.  
***NO\_COLOR\_PROCESSING*** No color processing.  
***NEAREST\_NEIGHBOR*** Fastest but lowest quality. Equivalent to FLYCAPTURE\_NEAREST\_NEIGHBOR↵  
 \_FAST in FlyCapture.  
***EDGE\_SENSING*** Weights surrounding pixels based on localized edge orientation.  
***HQ\_LINEAR*** Well-balanced speed and quality.  
***RIGOROUS*** Slowest but produces good results.  
***IPP*** Multi-threaded with similar results to edge sensing.  
***DIRECTIONAL\_FILTER*** Best quality but much faster than rigorous.  
***WEIGHTED\_DIRECTIONAL\_FILTER*** Weighted pixel average from different directions.

## 8.29.2.3 enum Error

[Spinnaker](#) enum definitions.

The error codes used in [Spinnaker](#). These codes are returned as part of [Spinnaker::Exception](#). The error codes in the range of -1000 to -1999 are reserved for exceptions that map directly to GenTL values. The error codes in the range of -2000 to -2999 are reserved for [GenICam](#) related errors. The error codes in the range of -3000 to -3999 are reserved for image processing related errors.

## Enumerator

***SPINNAKER\_ERR\_SUCCESS***  
***SPINNAKER\_ERR\_ERROR***  
***SPINNAKER\_ERR\_NOT\_INITIALIZED***  
***SPINNAKER\_ERR\_NOT\_IMPLEMENTED***  
***SPINNAKER\_ERR\_RESOURCE\_IN\_USE***  
***SPINNAKER\_ERR\_ACCESS\_DENIED***  
***SPINNAKER\_ERR\_INVALID\_HANDLE***  
***SPINNAKER\_ERR\_INVALID\_ID***  
***SPINNAKER\_ERR\_NO\_DATA***  
***SPINNAKER\_ERR\_INVALID\_PARAMETER***  
***SPINNAKER\_ERR\_IO***  
***SPINNAKER\_ERR\_TIMEOUT***  
***SPINNAKER\_ERR\_ABORT***  
***SPINNAKER\_ERR\_INVALID\_BUFFER***  
***SPINNAKER\_ERR\_NOT\_AVAILABLE***  
***SPINNAKER\_ERR\_INVALID\_ADDRESS***  
***SPINNAKER\_ERR\_BUFFER\_TOO\_SMALL***  
***SPINNAKER\_ERR\_INVALID\_INDEX***  
***SPINNAKER\_ERR\_PARSING\_CHUNK\_DATA***  
***SPINNAKER\_ERR\_INVALID\_VALUE***  
***SPINNAKER\_ERR\_RESOURCE\_EXHAUSTED***  
***SPINNAKER\_ERR\_OUT\_OF\_MEMORY***  
***SPINNAKER\_ERR\_BUSY***  
***GENICAM\_ERR\_INVALID\_ARGUMENT***  
***GENICAM\_ERR\_OUT\_OF\_RANGE***  
***GENICAM\_ERR\_PROPERTY***  
***GENICAM\_ERR\_RUN\_TIME***  
***GENICAM\_ERR\_LOGICAL***  
***GENICAM\_ERR\_ACCESS***  
***GENICAM\_ERR\_TIMEOUT***  
***GENICAM\_ERR\_DYNAMIC\_CAST***  
***GENICAM\_ERR\_GENERIC***  
***GENICAM\_ERR\_BAD\_ALLOCATION***  
***SPINNAKER\_ERR\_IM\_CONVERT***  
***SPINNAKER\_ERR\_IM\_COPY***  
***SPINNAKER\_ERR\_IM\_MALLOC***

***SPINNAKER\_ERR\_IM\_NOT\_SUPPORTED***  
***SPINNAKER\_ERR\_IM\_HISTOGRAM\_RANGE***  
***SPINNAKER\_ERR\_IM\_HISTOGRAM\_MEAN***  
***SPINNAKER\_ERR\_IM\_MIN\_MAX***  
***SPINNAKER\_ERR\_IM\_COLOR\_CONVERSION***  
***SPINNAKER\_ERR\_IM\_DECOMPRESSION***  
***SPINNAKER\_ERR\_CUSTOM\_ID***

#### 8.29.2.4 enum EventType

Event types in [Spinnaker](#).

See also

[Event::GetEventType\(\)](#)

Enumerator

***SPINNAKER\_EVENT\_ARRIVAL\_REMOVAL***  
***SPINNAKER\_EVENT\_DEVICE***  
***SPINNAKER\_EVENT\_DEVICE\_SPECIFIC***  
***SPINNAKER\_EVENT\_NEW\_BUFFER***  
***SPINNAKER\_EVENT\_LOGGING\_EVENT***  
***SPINNAKER\_EVENT\_UNKNOWN***

#### 8.29.2.5 enum HeatMapColor

Enumerator

***HEATMAP\_BLACK***  
***HEATMAP\_BLUE***  
***HEATMAP\_CYAN***  
***HEATMAP\_GREEN***  
***HEATMAP\_YELLOW***  
***HEATMAP\_RED***  
***HEATMAP\_WHITE***

#### 8.29.2.6 enum ImageFileFormat

File formats to be used for saving images to disk.

Enumerator

***FROM\_FILE\_EXT*** Determine file format from file extension.  
***PGM*** Portable gray map.  
***PPM*** Portable pixmap.  
***BMP*** Bitmap.  
***JPEG*** JPEG.  
***JPEG2000*** JPEG 2000.  
***TIFF*** Tagged image file format.  
***PNG*** Portable network graphics.  
***RAW*** Raw data.  
***JPEG12\_C*** 12 bit compressed JPEG data.  
***IMAGE\_FILE\_FORMAT\_FORCE\_32BITS***

## 8.29.2.7 enum ImageStatus

Status of images returned from `GetNextImage()` call.

## Enumerator

**IMAGE\_UNKNOWN\_ERROR** [Image](#) has an unknown error.

**IMAGE\_NO\_ERROR** [Image](#) is returned from `GetNextImage()` call without any errors.

**IMAGE\_CRC\_CHECK\_FAILED** [Image](#) failed CRC check.

**IMAGE\_DATA\_OVERFLOW** Received more data than the size of the image.

**IMAGE\_MISSING\_PACKETS** [Image](#) has missing packets.

**IMAGE\_LEADER\_BUFFER\_SIZE\_INCONSISTENT** [Image](#) leader is incomplete.

**IMAGE\_TRAILER\_BUFFER\_SIZE\_INCONSISTENT** [Image](#) trailer is incomplete.

**IMAGE\_PACKETID\_INCONSISTENT** [Image](#) has an inconsistent packet id.

**IMAGE\_MISSING\_LEADER** [Image](#) leader is missing.

**IMAGE\_MISSING\_TRAILER** [Image](#) trailer is missing.

**IMAGE\_DATA\_INCOMPLETE** [Image](#) data is incomplete.

**IMAGE\_INFO\_INCONSISTENT** [Image](#) info is corrupted.

**IMAGE\_CHUNK\_DATA\_INVALID** [Image](#) chunk data is invalid.

**IMAGE\_NO\_SYSTEM\_RESOURCES** [Image](#) cannot be processed due to lack of system resources.

## 8.29.2.8 enum PayloadTypeInfoIDs

## Enumerator

**PAYLOAD\_TYPE\_UNKNOWN**

**PAYLOAD\_TYPE\_IMAGE**

**PAYLOAD\_TYPE\_RAW\_DATA**

**PAYLOAD\_TYPE\_FILE**

**PAYLOAD\_TYPE\_CHUNK\_DATA**

**PAYLOAD\_TYPE\_JPEG**

**PAYLOAD\_TYPE\_JPEG2000**

**PAYLOAD\_TYPE\_H264**

**PAYLOAD\_TYPE\_CHUNK\_ONLY**

**PAYLOAD\_TYPE\_DEVICE\_SPECIFIC**

**PAYLOAD\_TYPE\_MULTI\_PART**

**PAYLOAD\_TYPE\_CUSTOM\_ID**

**PAYLOAD\_TYPE\_EXTENDED\_CHUNK**



## 8.29.2.9 enum PixelFormatIntType

Possible integer types and packing used in a pixel format.

Enumerator

***IntType\_UINT8***  
***IntType\_INT8***  
***IntType\_UINT10***  
***IntType\_UINT10p***  
***IntType\_UINT10P***  
***IntType\_UINT12***  
***IntType\_UINT12p***  
***IntType\_UINT12P***  
***IntType\_UINT14***  
***IntType\_UINT16***  
***IntType\_FLOAT32***  
***IntType\_UNKNOWN***

## 8.29.2.10 enum PixelFormatNamespaceID

This enum represents the namespace in which the TL specific pixel format resides.

This enum is returned from a captured image when calling [Image::GetTLPixelFormatNamespace\(\)](#). It can be used to interpret the raw pixel format returned from [Image::GetTLPixelFormat\(\)](#).

See also

[Image::GetTLPixelFormat\(\)](#)  
[Image::GetTLPixelFormatNamespace\(\)](#)

Enumerator

***SPINNAKER\_PIXELFORMAT\_NAMESPACE\_UNKNOWN***  
***SPINNAKER\_PIXELFORMAT\_NAMESPACE\_GEV***  
***SPINNAKER\_PIXELFORMAT\_NAMESPACE\_IIDC***  
***SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_16BIT***  
***SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_32BIT***  
***SPINNAKER\_PIXELFORMAT\_NAMESPACE\_CUSTOM\_ID***

## 8.29.2.11 enum PolarizationAlgorithm

## Enumerator

**NO\_POLARIZATION** No polarization.

**QUADRANT\_I0\_GRAYSCALE** Extracts a Mono8 pixel format image of the 0 degree of polarization. Polarization value pointer will be null.

**QUADRANT\_I45\_GRAYSCALE** Extracts a Mono8 pixel format image of the 45 degree of polarization. Polarization value pointer will be null.

**QUADRANT\_I90\_GRAYSCALE** Extracts a Mono8 pixel format image of the 90 degree of polarization. Polarization value pointer will be null.

**QUADRANT\_I135\_GRAYSCALE** Extracts a Mono8 pixel format image of the 135 degree of polarization. Polarization value pointer will be null.

**STOKES\_S0\_GRAYSCALE** Extracts a Mono8 pixel format Stokes' parameter image S0.

**STOKES\_S0\_HEATMAP** Extracts a BGra8 pixel format HeatMap representation of the Stokes' parameter image S0.

**STOKES\_S1\_GRAYSCALE** Extracts a Mono8 pixel format Stokes' parameter image S1.

**STOKES\_S1\_HEATMAP** Extracts a BGra8 pixel format HeatMap representation of the Stokes' parameter image S1.

**STOKES\_S2\_GRAYSCALE** Extracts a Mono8 pixel format Stokes' parameter image S2.

**STOKES\_S2\_HEATMAP** Extracts a BGra8 pixel format HeatMap representation of the Stokes' parameter image S2.

**DOLP\_GRAYSCALE** Extracts a Mono8 pixel format image representation of the DoLP (Degree of Linear Polarization).

**DOLP\_HEATMAP** Extracts a BGra8 pixel format HeatMap representation of the DoLP (Degree of Linear Polarization). Resulting polarization values are normalized between 0 and 1.

**AOP\_GRAYSCALE** Extracts a Mono8 pixel format image representation of the AoP (Angle of Polarization).

**AOP\_HEATMAP** Extracts a BGra8 pixel format HeatMap representation of the AoP (Angle of Polarization). Resulting polarization values are normalized between 0 and 1.

## 8.29.2.12 enum PolarizationResolution

## Enumerator

**QUARTER\_RESOLUTION** Quarter Resolution.

**FULL\_RESOLUTION** Full Resolution.

## 8.29.2.13 enum SpinnakerLogLevel

## log levels

## Enumerator

**LOG\_LEVEL\_OFF**

**LOG\_LEVEL\_FATAL**

**LOG\_LEVEL\_ALERT**

**LOG\_LEVEL\_CRIT**

**LOG\_LEVEL\_ERROR**

**LOG\_LEVEL\_WARN**

**LOG\_LEVEL\_NOTICE**

**LOG\_LEVEL\_INFO**

**LOG\_LEVEL\_DEBUG**

**LOG\_LEVEL\_NOTSET**

## 8.29.2.14 enum StatisticsChannel

Channels that allow statistics to be calculated.

Enumerator

***GREY***

***RED***

***GREEN***

***BLUE***

***HUE***

***SATURATION***

***LIGHTNESS***

***NUM\_STATISTICS\_CHANNELS***

## 8.30 Spinnaker Platform

Platform-specific header file for [Spinnaker](#).

Collaboration diagram for Spinnaker Platform:



### Macros

- `#define SPINNAKER_API_ABSTRACT /*nothing*/`
- `#define SPINNAKER_API __attribute__((visibility ("default")))`
- `#define SPINNAKER_LOCAL __attribute__((visibility ("hidden")))`

#### 8.30.1 Detailed Description

Platform-specific header file for [Spinnaker](#).

All the platform-specific code that is required by individual compilers to produce the appropriate code for each platform.

#### 8.30.2 Macro Definition Documentation

8.30.2.1 `#define SPINNAKER_API __attribute__((visibility ("default")))`

8.30.2.2 `#define SPINNAKER_API_ABSTRACT /*nothing*/`

8.30.2.3 `#define SPINNAKER_LOCAL __attribute__((visibility ("hidden")))`

## 8.31 Spinnaker Video Class

Collaboration diagram for Spinnaker Video Class:



### Classes

- class [SpinVideo](#)

*Provides the functionality for the user to record images to an AVI/MP4 file.*

#### 8.31.1 Detailed Description

## 8.32 Spinnaker Video Definitions

Definitions file for [Spinnaker](#) video recorder.

Collaboration diagram for Spinnaker Video Definitions:



Definitions file for [Spinnaker](#) video recorder.

## 8.33 System Class

Collaboration diagram for System Class:



### Classes

- class [System](#)

*The system object is used to retrieve the list of interfaces and cameras available.*

### 8.33.1 Detailed Description

## 8.34 SystemPtr Class

Collaboration diagram for SystemPtr Class:



### Classes

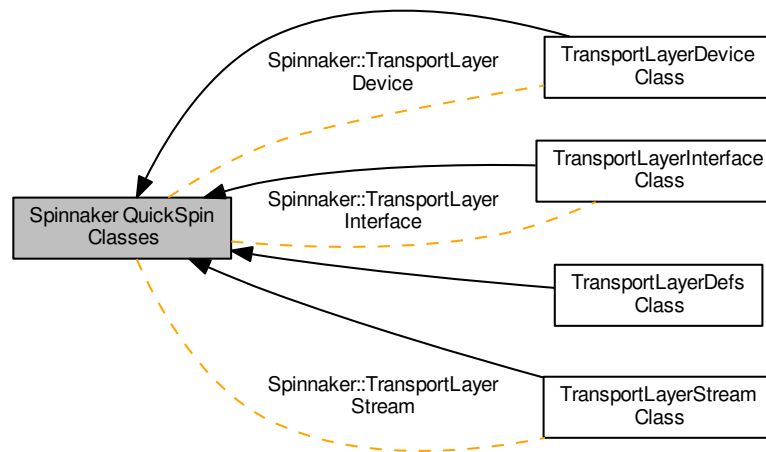
- class [SystemPtr](#)  
*A reference tracked pointer to a system object.*

#### 8.34.1 Detailed Description



## 8.35 Spinnaker QuickSpin Classes

Collaboration diagram for Spinnaker QuickSpin Classes:



### Modules

- [TransportLayerDefs Class](#)
- [TransportLayerDevice Class](#)
- [TransportLayerInterface Class](#)
- [TransportLayerStream Class](#)

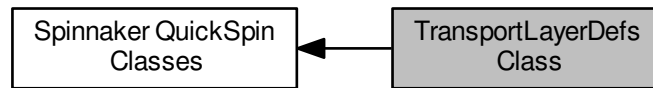
### Classes

- class [TransportLayerDevice](#)  
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*
- class [TransportLayerInterface](#)  
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*
- class [TransportLayerStream](#)  
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

#### 8.35.1 Detailed Description

## 8.36 TransportLayerDefs Class

Collaboration diagram for TransportLayerDefs Class:



### Enumerations

- enum [StreamTypeEnum](#) {  
[StreamType\\_Mixed](#),  
[StreamType\\_Custom](#),  
[StreamType\\_GEV](#),  
[StreamType\\_CL](#),  
[StreamType\\_IIDC](#),  
[StreamType\\_UVC](#),  
[StreamType\\_CXP](#),  
[StreamType\\_CLHS](#),  
[StreamType\\_U3V](#),  
[StreamType\\_ETHERNET](#),  
[StreamType\\_PCI](#),  
[NUMSTREAMTYPE](#) }

*The enum definitions for TL Device nodes from the transport layer .xml files.*

- enum [StreamDefaultBufferCountModeEnum](#) {  
[StreamDefaultBufferCountMode\\_Manual](#),  
[StreamDefaultBufferCountMode\\_Auto](#),  
[NUMSTREAMDEFAULTBUFFERCOUNTMODE](#) }
- enum [StreamBufferCountModeEnum](#) {  
[StreamBufferCountMode\\_Manual](#),  
[StreamBufferCountMode\\_Auto](#),  
[NUMSTREAMBUFFERCOUNTMODE](#) }
- enum [StreamBufferHandlingModeEnum](#) {  
[StreamBufferHandlingMode\\_OldestFirst](#),  
[StreamBufferHandlingMode\\_OldestFirstOverwrite](#),  
[StreamBufferHandlingMode\\_NewestFirst](#),  
[StreamBufferHandlingMode\\_NewestFirstOverwrite](#),  
[StreamBufferHandlingMode\\_NewestOnly](#),  
[NUMSTREAMBUFFERHANDLINGMODE](#) }
- enum [DeviceTypeEnum](#) {  
[DeviceType\\_Mixed](#),  
[DeviceType\\_Custom](#),  
[DeviceType\\_GEV](#),  
[DeviceType\\_CL](#),  
[DeviceType\\_IIDC](#),  
[DeviceType\\_UVC](#),  
[DeviceType\\_CXP](#),  
[DeviceType\\_CLHS](#),  
[DeviceType\\_U3V](#),  
[DeviceType\\_ETHERNET](#),  
[DeviceType\\_PCI](#),

```

    NUMDEVICETYPE }
    • enum DeviceAccessStatusEnum {
        DeviceAccessStatus_Unknown,
        DeviceAccessStatus_ReadWrite,
        DeviceAccessStatus_ReadOnly,
        DeviceAccessStatus_NoAccess,
        NUMDEVICEACCESSSTATUS }
    • enum GevCCPEnum {
        GevCCP_EnumEntry_GevCCP_OpenAccess,
        GevCCP_EnumEntry_GevCCP_ExclusiveAccess,
        GevCCP_EnumEntry_GevCCP_ControlAccess,
        NUMGEVCCP }
    • enum GUIXMLLocationEnum {
        GUIXMLLocation_Device,
        GUIXMLLocation_Host,
        NUMGUIXMLLOCATION }
    • enum GenICamXMLLocationEnum {
        GenICamXMLLocation_Device,
        GenICamXMLLocation_Host,
        NUMGENICAMXMLLOCATION }
    • enum DeviceEndiannessMechanismEnum {
        DeviceEndiannessMechanism_Legacy,
        DeviceEndiannessMechanism_Standard,
        NUMDEVICEENDIANESSMECHANISM }
    • enum DeviceCurrentSpeedEnum {
        DeviceCurrentSpeed_UnknownSpeed,
        DeviceCurrentSpeed_LowSpeed,
        DeviceCurrentSpeed_FullSpeed,
        DeviceCurrentSpeed_HighSpeed,
        DeviceCurrentSpeed_SuperSpeed,
        NUMDEVICECURRENTSPEED }
    • enum POEStatusEnum {
        POEStatus_NotSupported,
        POEStatus_PowerOff,
        POEStatus_PowerOn,
        NUMPOESTATUS }

```

### 8.36.1 Detailed Description

### 8.36.2 Enumeration Type Documentation

#### 8.36.2.1 enum DeviceAccessStatusEnum

< Gets the access status the transport layer Producer has on the device.

#### Enumerator

***DeviceAccessStatus\_Unknown*** Unknown status  
***DeviceAccessStatus\_ReadWrite*** Full access  
***DeviceAccessStatus\_ReadOnly*** Read-only access  
***DeviceAccessStatus\_NoAccess*** Non-available devices  
***NUMDEVICEACCESSSTATUS***

## 8.36.2.2 enum DeviceCurrentSpeedEnum

< The USB Speed that the device is currently operating at.

## Enumerator

**DeviceCurrentSpeed\_UnknownSpeed** Unknown-Speed.  
**DeviceCurrentSpeed\_LowSpeed** Low-Speed.  
**DeviceCurrentSpeed\_FullSpeed** Full-Speed.  
**DeviceCurrentSpeed\_HighSpeed** High-Speed.  
**DeviceCurrentSpeed\_SuperSpeed** Super-Speed.  
**NUMDEVICECURRENTSPEED**

## 8.36.2.3 enum DeviceEndiannessMechanismEnum

< Identifies the endianness handling mode.

## Enumerator

**DeviceEndiannessMechanism\_Legacy** Handling the device endianness according to [GenICam](#) Schema 1.0  
**DeviceEndiannessMechanism\_Standard** Handling the device endianness according to [GenICam](#) Schema 1.1 and later  
**NUMDEVICEENDIANESSMECHANISM**

## 8.36.2.4 enum DeviceTypeEnum

< Transport layer type of the device.

## Enumerator

**DeviceType\_Mixed** TL - Mixed  
**DeviceType\_Custom** TL - Custom  
**DeviceType\_GEV** TL - GEV  
**DeviceType\_CL** TL - CL  
**DeviceType\_IIDC** TL - IIDC  
**DeviceType\_UVC** TL - UVC  
**DeviceType\_CXP** TL - CXP  
**DeviceType\_CLHS** TL - CLHS  
**DeviceType\_U3V** TL - U3V  
**DeviceType\_ETHERNET** TL - ETHERNET  
**DeviceType\_PCI** TL - PCI  
**NUMDEVICETYPE**

## 8.36.2.5 enum GenICamXMLLocationEnum

< Sets the location to load [GenICam](#) XML.

## Enumerator

**GenICamXMLLocation\_Device** Load [GenICam](#) XML from device

**GenICamXMLLocation\_Host** Load [GenICam](#) XML from host

**NUMGENICAMXMLLOCATION**

## 8.36.2.6 enum GevCCPEnum

< Controls the device access privilege of an application.

## Enumerator

**GevCCP\_EnumEntry\_GevCCP\_OpenAccess** Open access privilege.

**GevCCP\_EnumEntry\_GevCCP\_ExclusiveAccess** Exclusive access privilege.

**GevCCP\_EnumEntry\_GevCCP\_ControlAccess** Control access privilege.

**NUMGEVCCP**

## 8.36.2.7 enum GUIXMLLocationEnum

< Sets the location to load GUI XML.

## Enumerator

**GUIXMLLocation\_Device** Load XML from device

**GUIXMLLocation\_Host** Load XML from host

**NUMGUIXMLLOCATION**

## 8.36.2.8 enum POEStatusEnum

< Reports and controls the interface's power over Ethernet status.

## Enumerator

**POEStatus\_NotSupported** Not Supported

**POEStatus\_PowerOff** Power is Off

**POEStatus\_PowerOn** Power is On

**NUMPOESTATUS**

## 8.36.2.9 enum StreamBufferCountModeEnum

< Controls access to setting the number of buffers used for the stream. Locked to Manual mode on 32-bit Windows due to memory constraints.

## Enumerator

**StreamBufferCountMode\_Manual** The number of buffers used for the stream are set by the user.

**StreamBufferCountMode\_Auto** The number of buffers used for the stream is automatically calculated based on the device frame rate.

**NUMSTREAMBUFFERCOUNTMODE**

## 8.36.2.10 enum StreamBufferHandlingModeEnum

< Available buffer handling modes of this data stream:

## Enumerator

**StreamBufferHandlingMode\_OldestFirst** The application always gets the buffer from the head of the output buffer queue (thus, the oldest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires.

**StreamBufferHandlingMode\_OldestFirstOverwrite** The application always gets the buffer from the head of the output buffer queue (thus, the oldest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires. If a new buffer arrives it will overwrite the existing buffer from the head of the queue (behaves like a circular buffer).

**StreamBufferHandlingMode\_NewestFirst** The application always gets the buffer from the tail of the output buffer queue (thus, the newest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires.

**StreamBufferHandlingMode\_NewestFirstOverwrite** DEPRECATED. This is replaced by NewestOnly.

**StreamBufferHandlingMode\_NewestOnly** The application always gets the latest completed buffer (the newest one). If the Output Buffer Queue is empty, the application waits for a newly acquired buffer until the timeout expires. This buffer handling mode is typically used in a live display GUI where it is important that there is no lag between camera and display.

**NUMSTREAMBUFFERHANDLINGMODE**

## 8.36.2.11 enum StreamDefaultBufferCountModeEnum

< DEPRECATED; Replaced by StreamBufferCountMode. Controls access to setting the number of buffers used for the stream. Locked to Manual mode on 32-bit Windows due to memory constraints.

## Enumerator

**StreamDefaultBufferCountMode\_Manual** DEPRECATED. The number of buffers used for the stream are set by the user.

**StreamDefaultBufferCountMode\_Auto** DEPRECATED. The number of buffers used for the stream is automatically calculated.

**NUMSTREAMDEFAULTBUFFERCOUNTMODE**

## 8.36.2.12 enum StreamTypeEnum

The enum definitions for TL Device nodes from the transport layer .xml files.

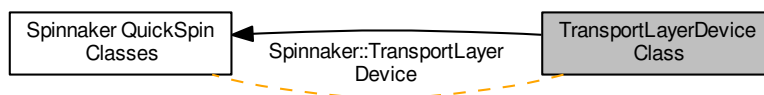
< Stream type of the device.

## Enumerator

***StreamType\_Mixed*** Stream Type - Mixed  
***StreamType\_Custom*** Stream Type - Custom  
***StreamType\_GEV*** Stream Type - GEV  
***StreamType\_CL*** Stream Type - CL  
***StreamType\_IIDC*** Stream Type - IIDC  
***StreamType\_UVC*** Stream Type - UVC  
***StreamType\_CXP*** Stream Type - CXP  
***StreamType\_CLHS*** Stream Type - CLHS  
***StreamType\_U3V*** Stream Type - U3V  
***StreamType\_ETHERNET*** Stream Type - ETHERNET  
***StreamType\_PCI*** Stream Type - PCI  
***NUMSTREAMTYPE***

## 8.37 TransportLayerDevice Class

Collaboration diagram for TransportLayerDevice Class:



### Classes

- class [TransportLayerDevice](#)

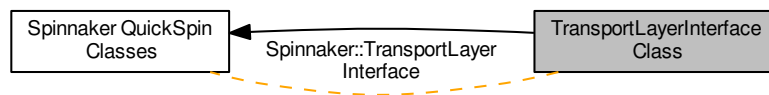
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

### 8.37.1 Detailed Description



## 8.38 TransportLayerInterface Class

Collaboration diagram for TransportLayerInterface Class:



### Classes

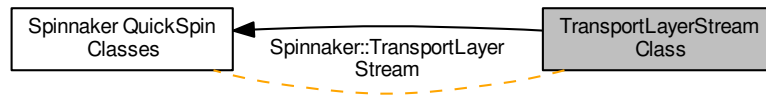
- class [TransportLayerInterface](#)

*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

### 8.38.1 Detailed Description

## 8.39 TransportLayerStream Class

Collaboration diagram for TransportLayerStream Class:



### Classes

- class [TransportLayerStream](#)

*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

### 8.39.1 Detailed Description

## 8.40 Camera Base Interface Class

Collaboration diagram for Camera Base Interface Class:



### Classes

- class [ICameraBase](#)

*The interface file for base class for the camera object.*

### 8.40.1 Detailed Description

## 8.41 IChunkData Class

Collaboration diagram for IChunkData Class:



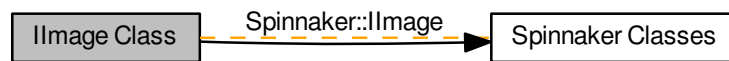
### Classes

- class [IChunkData](#)  
The [Interface](#) file for [ChunkData](#).

### 8.41.1 Detailed Description

## 8.42 IImage Class

Collaboration diagram for IImage Class:



### Classes

- class [IImage](#)  
*The interface file for [Image](#).*

### 8.42.1 Detailed Description

## 8.43 IImageStatistics Class

Collaboration diagram for IImageStatistics Class:



### Classes

- class [IImageStatistics](#)  
*The interface file for image statistics.*

#### 8.43.1 Detailed Description

## 8.44 IInterface Class

Collaboration diagram for IInterface Class:



### Classes

- class [IInterface](#)

*The interface file for [IInterface](#).*

#### 8.44.1 Detailed Description

## 8.45 IInterfaceList Class

Collaboration diagram for IInterfaceList Class:



### Classes

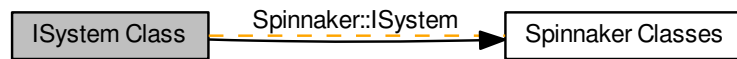
- class [IInterfaceList](#)  
*The interface file for [IInterfaceList](#) class.*

### 8.45.1 Detailed Description



## 8.46 ISystem Class

Collaboration diagram for ISystem Class:



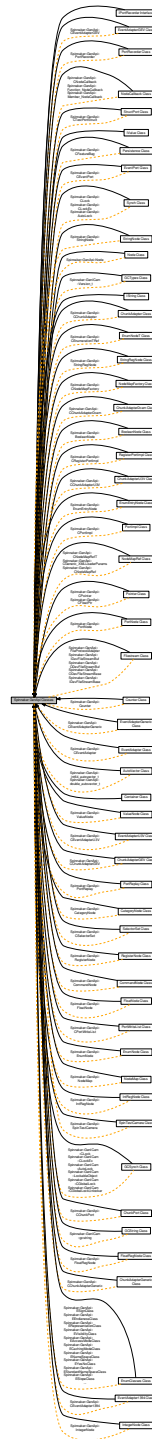
### Classes

- class [ISystem](#)  
*The interface file for [System](#).*

### 8.46.1 Detailed Description

## 8.47 Spinnaker GenApi Classes

Collaboration diagram for Spinnaker GenApi Classes:



### Modules

- [AutoVector Class](#)
- [BooleanNode Class](#)

- [CategoryNode Class](#)
- [ChunkAdapter Class](#)
- [ChunkAdapterDcam Class](#)
- [ChunkAdapterGeneric Class](#)
- [ChunkAdapterGEV Class](#)
- [ChunkPort Class](#)
- [CommandNode Class](#)
- [Container Class](#)
- [Counter Class](#)
- [EnumClasses Class](#)
- [EnumEntryNode Class](#)
- [EnumNode Class](#)
- [EnumNodeT Class](#)
- [EventAdapter Class](#)
- [EventAdapter1394 Class](#)
- [EventAdapterGeneric Class](#)
- [EventAdapterGEV Class](#)
- [EventAdapterU3V Class](#)
- [EventPort Class](#)
- [Filestream Class](#)
- [FloatNode Class](#)
- [FloatRegNode Class](#)
- [GCString Class](#)
- [GCSynch Class](#)
- [GCTypes Class](#)
- [IntegerNode Class](#)
- [IntRegNode Class](#)
- [IString Class](#)
- [IValue Class](#)
- [Node Class](#)
- [NodeCallback Class](#)
- [NodeMap Class](#)
- [NodeMapFactory Class](#)
- [NodeMapRef Class](#)
- [Persistence Class](#)
- [Pointer Class](#)
- [PortImpl Class](#)
- [PortNode Class](#)
- [PortRecorder Class](#)
- [PortReplay Class](#)
- [PortWriteList Class](#)
- [RegisterNode Class](#)
- [RegisterPortImpl Class](#)
- [SelectorSet Class](#)
- [SpinTestCamera Class](#)
- [StringNode Class](#)
- [StringRegNode Class](#)
- [StructPort Class](#)
- [Synch Class](#)
- [ValueNode Class](#)
- [ChunkAdapterU3V Class](#)
- [IPortRecorder Interface](#)

## Classes

- class [int64\\_autovector\\_t](#)  
*Vector of integers with reference counting.*
- class [double\\_autovector\\_t](#)  
*Vector of doubles with reference counting.*
- class [BooleanNode](#)  
*Interface for string properties.*
- class [CategoryNode](#)  
*Interface for string properties.*
- class [CChunkAdapter](#)  
*Connects a chunked buffer to a node map.*
- class [CChunkAdapterDcam](#)  
*Connects a chunked DCAM buffer to a node map.*
- class [CChunkAdapterGeneric](#)
- class [CChunkAdapterGEV](#)  
*Connects a chunked DCAM buffer to a node map.*
- class [CChunkAdapterU3V](#)  
*Connects a chunked U3V buffer to a node map.*
- class [CChunkPort](#)  
*Port attachable to a chunk in a buffer.*
- class [CommandNode](#)  
*Interface for string properties.*
- class [Counter](#)  
*Definition of a simple [Counter](#) class.*
- class [ESignClass](#)  
*Holds conversion methods for the sign enumeration.*
- class [EEndianessClass](#)  
*Holds conversion methods for the endianness enumeration.*
- class [ERepresentationClass](#)  
*Holds conversion methods for the representation enumeration.*
- class [EVisibilityClass](#)  
*Holds conversion methods for the visibility enumeration.*
- class [EAccessModeClass](#)  
*Holds conversion methods for the access mode enumeration.*
- class [ECachingModeClass](#)  
*Holds conversion methods for the caching mode enumeration.*
- class [ENamespaceClass](#)  
*Holds conversion methods for the namespace enumeration.*
- class [EYesNoClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [EStandardNameSpaceClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [ESlopeClass](#)  
*Holds conversion methods for the converter formulas.*
- class [EDisplayNotationClass](#)  
*Holds conversion methods for the notation type of floats.*
- class [EInputDirectionClass](#)  
*Holds conversion methods for the notation type of floats.*
- class [EGenApiSchemaVersionClass](#)  
*helper class converting [EGenApiSchemaVersion](#) from and to string*

- class [EnumEntryNode](#)  
*Interface for string properties.*
- class [EnumNode](#)  
*Interface for string properties.*
- class [CEnumerationTRef< EnumT >](#)  
*Interface for string properties.*
- class [CEventAdapter](#)  
*Delivers Events to ports.*
- class [CEventAdapter1394](#)  
*Distribute the events to the node map.*
- class [CEventAdapterGeneric](#)  
*Connects a generic event to a node map.*
- class [CEventAdapterGEV](#)  
*Connects a GigE [Event](#) to a node map.*
- class [CEventAdapterU3V](#)  
*Connects a U3V [Event](#) to a node map.*
- class [CEventPort](#)  
*Port attachable to an event.*
- class [FileProtocolAdapter](#)  
*Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.*
- class [IDevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBase< CharType, Traits >](#)
- class [IDevFileStreamBase< CharType, Traits >](#)
- class [FloatNode](#)  
*Interface for string properties.*
- class [FloatRegNode](#)  
*Interface for string properties.*
- class [gcstring](#)
- class [CLock](#)  
*A lock class.*
- class [CLockEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)
- class [LockableObject< Object >](#)  
*Instance-Lock for an object.*
- class [CGlobalLock](#)  
*Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)  
*Unlocks the global lock object on destruction.*
- struct [Version\\_t](#)  
*Version.*
- class [IntegerNode](#)  
*Interface for string properties.*
- class [IntRegNode](#)  
*Interface for string properties.*
- class [Node](#)  
*class common to all nodes*
- class [CNodeCallback](#)  
*callback body instance for INode pointers*
- class [Function\\_NodeCallback< Function >](#)

- Container for a function pointer.*

  - class [Member\\_NodeCallback](#)< Client, Member >
- Container for a member function pointer.*

  - class [NodeMap](#)
- Smart pointer template for NodeMaps with create function.*

  - class [CNodeMapFactory](#)
- The node map factory is used for creating node maps from camera description files.*

  - class [CNodeMapRefT](#)< TCameraParams >
- Smartpointer template for NodeMaps with create function.*

  - class [CGeneric\\_XMLLoaderParams](#)
- Empty base class used by class [CNodeMapRef](#) as generic template argument.*

  - class [CNodeMapRef](#)
- Smartpointer for NodeMaps with create function.*

  - class [CFeatureBag](#)
- Bag holding streamable features of a nodetree.*

  - class [CPointer](#)< T, B >
- Encapsulates a [GenApi](#) pointer dealing with the dynamic\_cast automatically.*

  - class [CFloatPtr](#)
- SmartPointer for IFloat interface pointer.*

  - class [CPortImpl](#)
- Standard implementation for a port.*

  - class [PortNode](#)
- [Interface](#) for value properties.*

  - class [PortRecorder](#)
- [Interface](#) for recording write commands on a port.*

  - class [PortReplay](#)
- [Interface](#) for replaying write commands on a port.*

  - class [CPortWriteList](#)
- Container holding a list of port write commands.*

  - class [RegisterNode](#)
- [Interface](#) for string properties.*

  - class [CRegisterPortImpl](#)
- Standard implementation for a port using a register based transport layer.*

  - class [CSelectorSet](#)
- The set of selectors selecting a given node.*

  - class [SpinTestCamera](#)
- [Interface](#) for string properties.*

  - class [StringNode](#)
- [Interface](#) for string properties.*

  - class [StringRegNode](#)
- [Interface](#) for string properties.*

  - class [CTestPortStruct](#)< CDataStruct >
- Implements a register spaces based on a C++ struct.*

  - class [CLock](#)
- A lock class.*

  - class [CLockEx](#)
- This class is for testing purposes only.*

  - class [AutoLock](#)
- [Interface](#) for value properties.*

  - class [ValueNode](#)

## Typedefs

- typedef Node [CNodeRef](#)
- typedef Node [CSelectorRef](#)
- typedef NodeMap [CNodeMapRef](#)

## Functions

- [SPINNAKER\\_API](#) IDestroy \* [CastToIDestroy](#) (INodeMap \*pNodeMap)  
*makes sure the dynamic\_cast operator is implemented in the DLL (due to a Linux bug)*
- template<class TCameraParams >  
void [\\_LoadXMLFromFile](#) (const GenICam::gcstring &FileName)
- template<class TCameraParams >  
void [\\_LoadXMLFromZIPFile](#) (const GenICam::gcstring &ZipFileName)
- template<class TCameraParams >  
void [\\_LoadXMLFromFileInject](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)
- template<class TCameraParams >  
void [\\_LoadXMLFromString](#) (const GenICam::gcstring &XMLData)
- template<class TCameraParams >  
void [\\_LoadXMLFromZIPData](#) (const void \*zipData, size\_t zipSize)
- template<class TCameraParams >  
void [\\_LoadXMLFromStringInject](#) (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)
- template<class TCameraParams >  
void [\\_GetSupportedSchemaVersions](#) (GenICam::gcstring\_vector &SchemaVersions)
- template<class TCameraParams >  
GenICam::gcstring [\\_GetDeviceName](#) ()
- template<class TCameraParams >  
void [\\_Poll](#) (int64\_t ElapsedTime)
- template<class TCameraParams >  
void [\\_GetNodes](#) (NodeList\_t &Nodes)
- template<class TCameraParams >  
INode \* [\\_GetNode](#) (const GenICam::gcstring &key)
- template<class TCameraParams >  
void [\\_InvalidateNodes](#) ()
- template<class TCameraParams >  
bool [\\_Connect](#) (IPort \*pPort, const GenICam::gcstring &PortName)
- template<class TCameraParams >  
bool [\\_Connect](#) (IPort \*pPort)
- template<class TCameraParams >  
bool [\\_ClearXMLCache](#) ()
- [SPINNAKER\\_API](#) std::istream & [EatComments](#) (std::istream &is)  
*Helper function ignoring lines starting with comment character '#'.*
- [SPINNAKER\\_API](#) std::istream & [operator>>](#) (std::istream &is, CFeatureBag &FeatureBag)  
*Reads in persistent data from a stream.*
- [SPINNAKER\\_API](#) std::ostream & [operator<<](#) (std::ostream &os, const CFeatureBag &FeatureBag)  
*writes out persistent data to a stream*
- [CNodeMapRefT](#) (const GenICam::gcstring &DeviceName="Device")  
*Constructor.*
- [CNodeMapRefT](#) (INodeMap \*pNodeMap, const GenICam::gcstring &DeviceName="Device")  
*Constructor.*
- [CNodeMapRefT](#) (const CNodeMapRefT &Them)  
*Copy constructor.*

- CNodeMapRefT & [operator=](#) (INodeMap \*pNodeMap)  
*Assignment of an INodeMap\*.*
- CNodeMapRefT & [operator=](#) (const CNodeMapRefT &Them)  
*Assignment.*
- virtual [~CNodeMapRefT](#) ()  
*Destructor.*
- void [\\_Destroy](#) ()  
*Destroys the node map.*

### 8.47.1 Detailed Description

### 8.47.2 Typedef Documentation

#### 8.47.2.1 typedef NodeMap CNodeMapRef

#### 8.47.2.2 typedef Node CNodeRef

#### 8.47.2.3 typedef Node CSelectorRef

### 8.47.3 Function Documentation

#### 8.47.3.1 bool Spinnaker::GenApi::\_ClearXMLCache ( ) [inline]

#### 8.47.3.2 bool Spinnaker::GenApi::\_Connect ( IPort \* pPort, const GenICam::gcstring & PortName ) [inline]

#### 8.47.3.3 bool Spinnaker::GenApi::\_Connect ( IPort \* pPort ) [inline]

#### 8.47.3.4 void \_Destroy ( ) [inline]

Destroys the node map.

#### 8.47.3.5 GenICam::gcstring Spinnaker::GenApi::\_GetDeviceName ( ) [inline]

#### 8.47.3.6 INode\* Spinnaker::GenApi::\_GetNode ( const GenICam::gcstring & key ) [inline]

#### 8.47.3.7 void Spinnaker::GenApi::\_GetNodes ( NodeList\_t & Nodes ) [inline]

#### 8.47.3.8 void Spinnaker::GenApi::\_GetSupportedSchemaVersions ( GenICam::gcstring\_vector & SchemaVersions ) [inline]

#### 8.47.3.9 void Spinnaker::GenApi::\_InvalidateNodes ( ) [inline]

#### 8.47.3.10 void Spinnaker::GenApi::\_LoadXMLFromFile ( const GenICam::gcstring & FileName ) [inline]



8.47.3.11 `void Spinnaker::GenApi::_LoadXMLFromFileInject ( const GenICam::gcstring & TargetFileName, const GenICam::gcstring & InjectFileName ) [inline]`

8.47.3.12 `void Spinnaker::GenApi::_LoadXMLFromString ( const GenICam::gcstring & XMLData ) [inline]`

8.47.3.13 `void Spinnaker::GenApi::_LoadXMLFromStringInject ( const GenICam::gcstring & TargetXMLData, const GenICam::gcstring & InjectXMLData ) [inline]`

8.47.3.14 `void Spinnaker::GenApi::_LoadXMLFromZIPData ( const void * zipData, size_t zipSize ) [inline]`

8.47.3.15 `void Spinnaker::GenApi::_LoadXMLFromZIPFile ( const GenICam::gcstring & ZipFileName ) [inline]`

8.47.3.16 `void Spinnaker::GenApi::_Poll ( int64_t ElapsedTime ) [inline]`

8.47.3.17 `SPINNAKER_API IDestroy* Spinnaker::GenApi::CastToIDestroy ( INodeMap * pNodeMap )`

makes sure the `dynamic_cast` operator is implemented in the DLL (due to a Linux bug)

8.47.3.18 `CNodeMapRefT ( const GenICam::gcstring & DeviceName = "Device" ) [inline]`

Constructor.

8.47.3.19 `CNodeMapRefT ( INodeMap * pNodeMap, const GenICam::gcstring & DeviceName = "Device" ) [inline]`

Constructor.

8.47.3.20 `CNodeMapRefT ( const CNodeMapRefT< TCameraParams > & Them )`

Copy constructor.

8.47.3.21 `SPINNAKER_API std::istream& Spinnaker::GenApi::EatComments ( std::istream & is )`

Helper function ignoring lines starting with comment character '#'.

8.47.3.22 `SPINNAKER_API std::ostream& Spinnaker::GenApi::operator<< ( std::ostream & os, const CFeatureBag & FeatureBag )`

writes out persistent data to a stream

8.47.3.23 `CNodeMapRefT< TCameraParams > & operator= ( const CNodeMapRefT< TCameraParams > & Them )`

Assignment.

8.47.3.24 `CNodeMapRefT< TCameraParams > & operator= ( INodeMap * pNodeMap )`

Assignment of an INodeMap\*.

8.47.3.25 `SPINNAKER_API std::istream& Spinnaker::GenApi::operator>> ( std::istream & is, CFeatureBag & FeatureBag )`

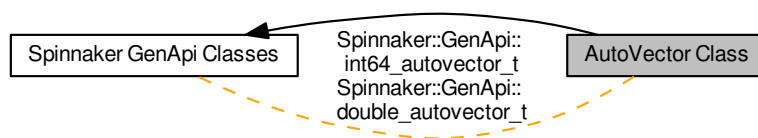
Reads in persistent data from a stream.

8.47.3.26 `~CNodeMapRefT ( ) [inline],[virtual]`

Destructor.

## 8.48 AutoVector Class

Collaboration diagram for AutoVector Class:



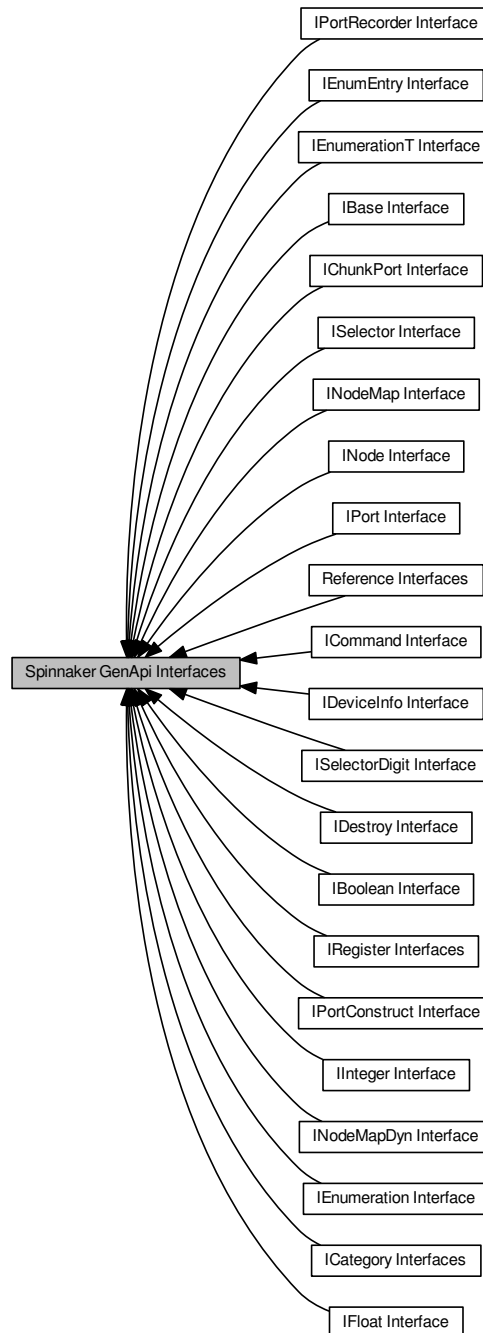
### Classes

- class [int64\\_autovector\\_t](#)  
*Vector of integers with reference counting.*
- class [double\\_autovector\\_t](#)  
*Vector of doubles with reference counting.*

### 8.48.1 Detailed Description

## 8.49 Spinnaker GenApi Interfaces

Collaboration diagram for Spinnaker GenApi Interfaces:



### Modules

- [IBase Interface](#)
- [IBoolean Interface](#)

- [ICategory Interfaces](#)
- [IChunkPort Interface](#)
- [ICommand Interface](#)
- [IDestroy Interface](#)
- [IDeviceInfo Interface](#)
- [IEnumEntry Interface](#)
- [IEnumeration Interface](#)
- [IEnumerationT Interface](#)
- [IFloat Interface](#)
- [IInteger Interface](#)
- [INode Interface](#)
- [INodeMap Interface](#)
- [INodeMapDyn Interface](#)
- [IPort Interface](#)
- [IPortConstruct Interface](#)
- [IPortRecorder Interface](#)
- [IRegister Interfaces](#)
- [ISelector Interface](#)
- [ISelectorDigit Interface](#)
- [Reference Interfaces](#)

## Typedefs

- typedef node\_vector [NodeList\\_t](#)  
*a list of node references*
- typedef intptr\_t [CallbackHandleType](#)  
*the callback handle for nodes*

### 8.49.1 Detailed Description

### 8.49.2 Typedef Documentation

#### 8.49.2.1 typedef intptr\_t CallbackHandleType

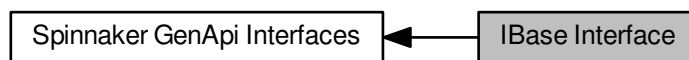
the callback handle for nodes

#### 8.49.2.2 typedef node\_vector NodeList\_t

a list of node references

## 8.50 IBase Interface

Collaboration diagram for IBase Interface:



### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IBase](#)  
*Base interface common to all nodes.*

### 8.50.1 Detailed Description

### 8.50.2 Variable Documentation

#### 8.50.2.1 interface SPINNAKER\_API\_ABSTRACT IBase

##### Initial value:

```
{  
    virtual EAccessMode GetAccessMode() const = 0
```

Base interface common to all nodes.

## 8.51 BooleanNode Class

Collaboration diagram for BooleanNode Class:



### Classes

- class [BooleanNode](#)  
*[Interface](#) for string properties.*

### Typedefs

- typedef BooleanNode [CBooleanRef](#)

#### 8.51.1 Detailed Description

#### 8.51.2 Typedef Documentation

##### 8.51.2.1 typedef BooleanNode CBooleanRef

## 8.52 CategoryNode Class

Collaboration diagram for CategoryNode Class:



### Classes

- class [CategoryNode](#)  
*Interface for string properties.*

### Typedefs

- typedef CategoryNode [CCategoryRef](#)

#### 8.52.1 Detailed Description

#### 8.52.2 Typedef Documentation

##### 8.52.2.1 typedef CategoryNode CCategoryRef



## 8.53 ChunkAdapter Class

Collaboration diagram for ChunkAdapter Class:



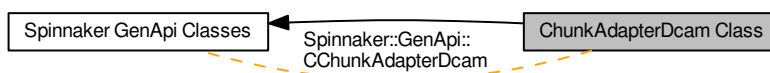
### Classes

- class [CChunkAdapter](#)  
*Connects a chunked buffer to a node map.*

#### 8.53.1 Detailed Description

## 8.54 ChunkAdapterDcam Class

Collaboration diagram for ChunkAdapterDcam Class:



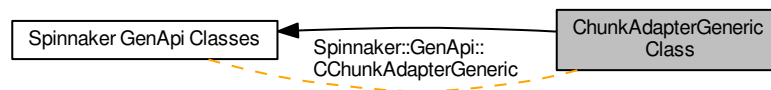
### Classes

- class [CChunkAdapterDcam](#)  
*Connects a chunked DCAM buffer to a node map.*

#### 8.54.1 Detailed Description

## 8.55 ChunkAdapterGeneric Class

Collaboration diagram for ChunkAdapterGeneric Class:



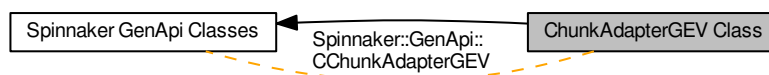
### Classes

- class [CChunkAdapterGeneric](#)

### 8.55.1 Detailed Description

## 8.56 ChunkAdapterGEV Class

Collaboration diagram for ChunkAdapterGEV Class:



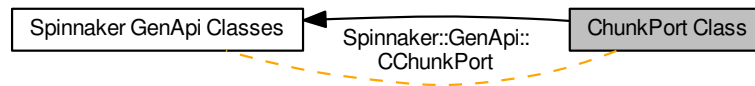
### Classes

- class [CChunkAdapterGEV](#)  
*Connects a chunked DCAM buffer to a node map.*

#### 8.56.1 Detailed Description

## 8.57 ChunkPort Class

Collaboration diagram for ChunkPort Class:



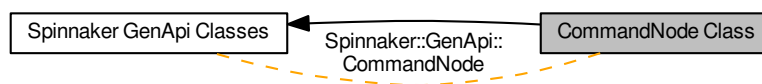
### Classes

- class [CChunkPort](#)  
*Port attachable to a chunk in a buffer.*

#### 8.57.1 Detailed Description

## 8.58 CommandNode Class

Collaboration diagram for CommandNode Class:



### Classes

- class [CommandNode](#)  
*[Interface](#) for string properties.*

### Typedefs

- typedef CommandNode [CCommandRef](#)

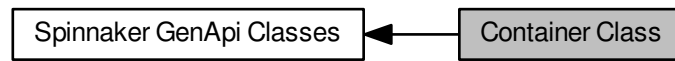
#### 8.58.1 Detailed Description

#### 8.58.2 Typedef Documentation

##### 8.58.2.1 typedef CommandNode CCommandRef

## 8.59 Container Class

Collaboration diagram for Container Class:



## 8.60 Counter Class

Collaboration diagram for Counter Class:



### Classes

- class [Counter](#)

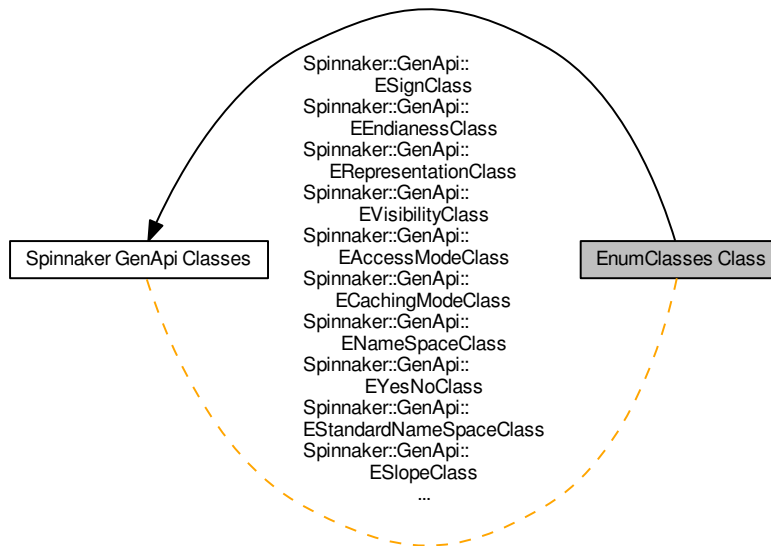
*Definition of a simple [Counter](#) class.*

### 8.60.1 Detailed Description



## 8.61 EnumClasses Class

Collaboration diagram for EnumClasses Class:



### Classes

- class [ESignClass](#)  
*Holds conversion methods for the sign enumeration.*
- class [EEndiannessClass](#)  
*Holds conversion methods for the endianness enumeration.*
- class [ERepresentationClass](#)  
*Holds conversion methods for the representation enumeration.*
- class [EVisibilityClass](#)  
*Holds conversion methods for the visibility enumeration.*
- class [EAccessModeClass](#)  
*Holds conversion methods for the access mode enumeration.*
- class [ECachingModeClass](#)  
*Holds conversion methods for the caching mode enumeration.*
- class [ENamespaceClass](#)  
*Holds conversion methods for the namespace enumeration.*
- class [EYesNoClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [EStandardNameSpaceClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [ESlopeClass](#)  
*Holds conversion methods for the converter formulas.*
- class [EDisplayNotationClass](#)  
*Holds conversion methods for the notation type of floats.*
- class [EInputDirectionClass](#)  
*Holds conversion methods for the notation type of floats.*
- class [EGenApiSchemaVersionClass](#)  
*helper class converting EGenApiSchemaVersion from and to string*

### 8.61.1 Detailed Description

## 8.62 EnumEntryNode Class

Collaboration diagram for EnumEntryNode Class:



### Classes

- class [EnumEntryNode](#)  
*Interface for string properties.*

### Typedefs

- typedef EnumEntryNode [CEnumEntryRef](#)

#### 8.62.1 Detailed Description

#### 8.62.2 Typedef Documentation

##### 8.62.2.1 typedef EnumEntryNode CEnumEntryRef

## 8.63 EnumNode Class

Collaboration diagram for EnumNode Class:



### Classes

- class [EnumNode](#)  
*[Interface](#) for string properties.*

### Typedefs

- typedef EnumNode [CEnumerationRef](#)

#### 8.63.1 Detailed Description

#### 8.63.2 Typedef Documentation

##### 8.63.2.1 typedef EnumNode CEnumerationRef

## 8.64 EnumNodeT Class

Collaboration diagram for EnumNodeT Class:



### Classes

- class [CEnumerationTRef](#)< [EnumT](#) >  
*Interface for string properties.*

### 8.64.1 Detailed Description

## 8.65 EventAdapter Class

Collaboration diagram for EventAdapter Class:



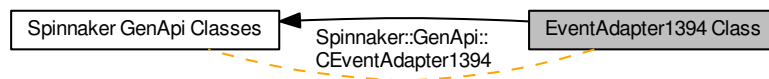
### Classes

- class [CEventAdapter](#)  
*Delivers Events to ports.*

### 8.65.1 Detailed Description

## 8.66 EventAdapter1394 Class

Collaboration diagram for EventAdapter1394 Class:



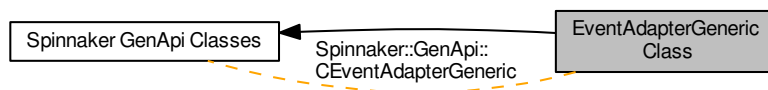
### Classes

- class [CEventAdapter1394](#)  
*Distribute the events to the node map.*

### 8.66.1 Detailed Description

## 8.67 EventAdapterGeneric Class

Collaboration diagram for EventAdapterGeneric Class:



### Classes

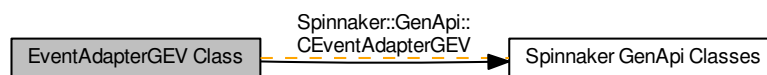
- class [CEventAdapterGeneric](#)  
*Connects a generic event to a node map.*

### 8.67.1 Detailed Description



## 8.68 EventAdapterGEV Class

Collaboration diagram for EventAdapterGEV Class:



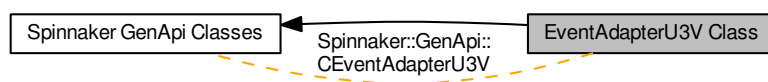
### Classes

- class [CEventAdapterGEV](#)  
*Connects a GigE [Event](#) to a node map.*

#### 8.68.1 Detailed Description

## 8.69 EventAdapterU3V Class

Collaboration diagram for EventAdapterU3V Class:



### Classes

- class [CEventAdapterU3V](#)  
*Connects a U3V [Event](#) to a node map.*

### 8.69.1 Detailed Description

## 8.70 EventPort Class

Collaboration diagram for EventPort Class:



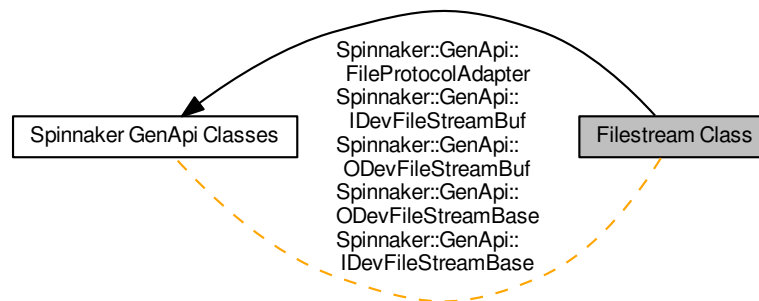
### Classes

- class [CEventPort](#)  
*Port attachable to an event.*

### 8.70.1 Detailed Description

## 8.71 Filestream Class

Collaboration diagram for Filestream Class:



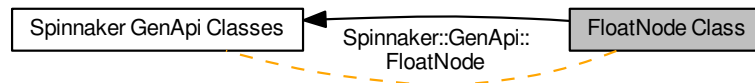
### Classes

- class [FileProtocolAdapter](#)  
Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.
- class [IDevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBase< CharType, Traits >](#)
- class [IDevFileStreamBase< CharType, Traits >](#)

### 8.71.1 Detailed Description

## 8.72 FloatNode Class

Collaboration diagram for FloatNode Class:



### Classes

- class [FloatNode](#)  
*Interface for string properties.*

### Typedefs

- typedef FloatNode [CFloatRef](#)

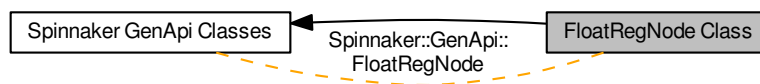
#### 8.72.1 Detailed Description

#### 8.72.2 Typedef Documentation

##### 8.72.2.1 typedef FloatNode CFloatRef

## 8.73 FloatRegNode Class

Collaboration diagram for FloatRegNode Class:



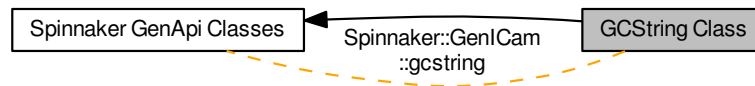
### Classes

- class [FloatRegNode](#)  
*Interface for string properties.*

#### 8.73.1 Detailed Description

## 8.74 GCString Class

Collaboration diagram for GCString Class:



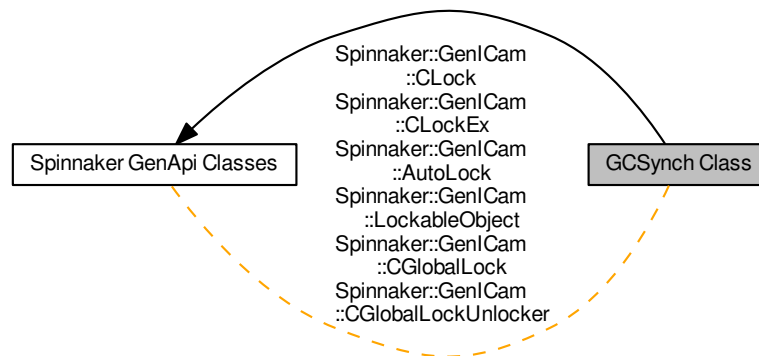
### Classes

- class [gcstring](#)

### 8.74.1 Detailed Description

## 8.75 GCSynch Class

Collaboration diagram for GCSynch Class:



### Classes

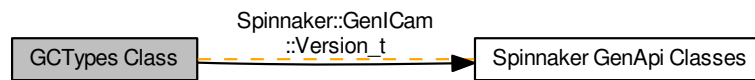
- class [CLock](#)  
*A lock class.*
- class [CLockEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)
- class [LockableObject< Object >](#)  
*Instance-Lock for an object.*
- class [CGlobalLock](#)  
*Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)  
*Unlocks the global lock object on destruction.*

### 8.75.1 Detailed Description



## 8.76 GCTypes Class

Collaboration diagram for GCTypes Class:



### Classes

- struct [Version\\_t](#)  
*Version.*

### Typedefs

- typedef float [float32\\_t](#)  
*32 bit floating point*
- typedef double [float64\\_t](#)  
*64 bit floating point*

#### 8.76.1 Detailed Description

#### 8.76.2 Typedef Documentation

##### 8.76.2.1 typedef float [float32\\_t](#)

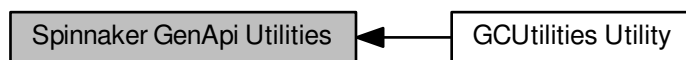
32 bit floating point

##### 8.76.2.2 typedef double [float64\\_t](#)

64 bit floating point

## 8.77 Spinnaker GenApi Utilities

Collaboration diagram for Spinnaker GenApi Utilities:



### Modules

- [GCUtilities Utility](#)

### 8.77.1 Detailed Description

## 8.78 GCUtilities Utility

Collaboration diagram for GCUtilities Utility:



### Functions

- `template<typename Td , typename Ts >`  
`Td INTEGRAL_CAST2 (Ts s)`  
*This verifies at runtime if there was no loss of data if an type Ts (e.g.*
- `template<typename T >`  
`T INTEGRAL_CAST (int64_t ll)`  
*This verifies at runtime if there was no loss of data if an int64\_t was downcast to type T (e.g.*
- `SPINNAKER_API bool DoesEnvironmentVariableExist (const Spinnaker::GenICam::gcstring &VariableName)`  
*Returns true if an environment variable exists.*
- `SPINNAKER_API gcstring GetValueOfEnvironmentVariable (const gcstring &VariableName)`  
*Retrieve the value of an environment variable.*
- `SPINNAKER_API bool GetValueOfEnvironmentVariable (const gcstring &VariableName, gcstring &VariableContent)`  
*Retrieve the value of an environment variable.*
- `SPINNAKER_API gcstring UriEncode (const gcstring &Input)`  
*Converts \ to / and replaces all unsafe characters by their xx equivalent.*
- `SPINNAKER_API gcstring UriDecode (const gcstring &Input)`  
*Replaces xx escapes by their char equivalent.*
- `SPINNAKER_API void ReplaceEnvironmentVariables (gcstring &Buffer, bool ReplaceBlankBy20=false)`  
*Replaces in a string and replace ' ' with %20.*
- `SPINNAKER_API gcstring GetGenICamCacheFolder (void)`  
*Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling SetGenICamCacheFolder().*
- `SPINNAKER_API gcstring GetGenICamLogConfig (void)`  
*Retrieve the path of the GenICam logging properties file.*
- `SPINNAKER_API gcstring GetGenICamCLProtocolFolder (void)`  
*Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling SetGenICamCLProtocolFolder().*
- `SPINNAKER_API void SetGenICamCacheFolder (const gcstring &path)`  
*Stores the path of the GenICam cache folder.*
- `SPINNAKER_API void SetGenICamLogConfig (const gcstring &path)`  
*Stores the path of the GenICam logging properties file.*
- `SPINNAKER_API void SetGenICamCLProtocolFolder (const gcstring &path)`  
*Stores the path of the CLProtocol folder.*
- `SPINNAKER_API void Tokenize (const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")`

*splits str input string into a list of tokens using the delimiter*

- **SPINNAKER\_API** void [GetFiles](#) (const gcstring &FileTemplate, gcstring\_vector &FileNames, const bool DirectoriesOnly=false)

*Gets a list of files or directories matching a given FileTemplate.*

- **SPINNAKER\_API** gcstring [GetModulePathFromFunction](#) (void \*pFunction)

*Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.*

### 8.78.1 Detailed Description

### 8.78.2 Function Documentation

#### 8.78.2.1 **SPINNAKER\_API** bool Spinnaker::GenICam::DoesEnvironmentVariableExist ( const Spinnaker::GenICam::gcstring & VariableName )

Returns true if an environment variable exists.

#### 8.78.2.2 **SPINNAKER\_API** void Spinnaker::GenICam::GetFiles ( const gcstring & FileTemplate, gcstring\_vector & FileNames, const bool DirectoriesOnly = false )

Gets a list of files or directories matching a given FileTemplate.

##### Parameters

<i>FileNames</i>	The file template. Can contain environment variables.
<i>DirectoriesOnly</i>	A list of files matching the file template

#### 8.78.2.3 **SPINNAKER\_API** gcstring Spinnaker::GenICam::GetGenICamCacheFolder ( void )

Retrieve the path of the [GenICam](#) cache folder The path to the cache folder can be stored by calling [SetGenICamCacheFolder\(\)](#).

If [GetGenICamCacheFolder\(\)](#) is called before [SetGenICamCacheFolder\(\)](#), it will return the value of environment variable GENICAM\_CACHE\_Vx\_y. If this environment variable does not exist, an exception will be thrown.

#### 8.78.2.4 **SPINNAKER\_API** gcstring Spinnaker::GenICam::GetGenICamCLProtocolFolder ( void )

Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling [SetGenICamCLProtocolFolder\(\)](#).

If [GetGenICamCLProtocolFolder\(\)](#) is called before [SetGenICamCLProtocolFolder\(\)](#), it will return the value of environment variable GENICAM\_CLPROTOCOL. If this environment variable does not exist, an exception will be thrown.

**8.78.2.5 SPINNAKER\_API** gcstring Spinnaker::GenICam::GetGenICamLogConfig ( void )

Retrieve the path of the [GenICam](#) logging properties file.

The path to the logging properties file can be stored by calling [SetGenICamLogConfig\(\)](#). If [GetGenICamLogConfig\(\)](#) is called before [SetGenICamLogConfig\(\)](#), it will return the value of environment variable GENICAM\_LOG\_CONFIG\_Vx\_y. If this environment variable does not exist, an exception will be thrown.

**8.78.2.6 SPINNAKER\_API** gcstring Spinnaker::GenICam::GetModulePathFromFunction ( void \* pFunction )

Gets the full path to the module (DLL/SO) containing the given *pFunction*; empty string if not found.

true = only subdirectories (ex . and ..) are retrieved; false = only files are retrieved

**8.78.2.7 SPINNAKER\_API** gcstring Spinnaker::GenICam::GetValueOfEnvironmentVariable ( const gcstring & VariableName )

Retrieve the value of an environment variable.

**Exceptions**

<i>runtime_exception</i>	if not found
--------------------------	--------------

**8.78.2.8 SPINNAKER\_API** bool Spinnaker::GenICam::GetValueOfEnvironmentVariable ( const gcstring & VariableName, gcstring & VariableContent )

Retrieve the value of an environment variable.

**Returns**

true if environment variable was found, otherwise false

**8.78.2.9** T Spinnaker::GenICam::INTEGRAL\_CAST ( int64\_t // ) [inline]

This verifies at runtime if there was no loss of data if an int64\_t was downcast to type T (e.g.

int32\_t)

**8.78.2.10** Td Spinnaker::GenICam::INTEGRAL\_CAST2 ( Ts s ) [inline]

This verifies at runtime if there was no loss of data if an type Ts (e.g.

int64t) was downcast to type Td (e.g. int32\_t)

**8.78.2.11 SPINNAKER\_API** void Spinnaker::GenlCam::ReplaceEnvironmentVariables ( gcstring & *Buffer*, bool *ReplaceBlankBy20 = false* )

Replaces in a string and replace ' ' with %20.

**8.78.2.12 SPINNAKER\_API** void Spinnaker::GenlCam::SetGenlCamCacheFolder ( const gcstring & *path* )

Stores the path of the [GenlCam](#) cache folder.

**8.78.2.13 SPINNAKER\_API** void Spinnaker::GenlCam::SetGenlCamCLProtocolFolder ( const gcstring & *path* )

Stores the path of the CLProtocol folder.

**8.78.2.14 SPINNAKER\_API** void Spinnaker::GenlCam::SetGenlCamLogConfig ( const gcstring & *path* )

Stores the path of the [GenlCam](#) logging properties file.

**8.78.2.15 SPINNAKER\_API** void Spinnaker::GenlCam::Tokenize ( const gcstring & *str*, gcstring\_vector & *tokens*, const gcstring & *delimiters* = " " )

splits str input string into a list of tokens using the delimiter

#### Parameters

<i>str</i>	string to be split
<i>tokens</i>	result of the splitting operation
<i>delimiters</i>	delimiters for the splitting

**8.78.2.16 SPINNAKER\_API** gcstring Spinnaker::GenlCam::UrlDecode ( const gcstring & *Input* )

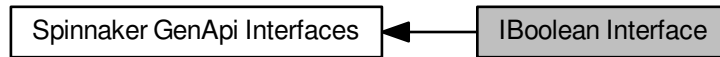
Replaces xx escapes by their char equivalent.

**8.78.2.17 SPINNAKER\_API** gcstring Spinnaker::GenlCam::UrlEncode ( const gcstring & *Input* )

Converts \ to / and replaces all unsave characters by their xx equivalent.

## 8.79 IBoolean Interface

Collaboration diagram for IBoolean Interface:



### Functions

- virtual void `operator=` (bool Value)  
*Set node value.*
- virtual bool `GetValue` (bool Verify=false, bool IgnoreCache=false) const =0  
*Get node value.*
- virtual bool `operator()` () const  
*Get node value.*

### Variables

- `interface SPINNAKER_API_ABSTRACT IBoolean`  
*Interface for Boolean properties.*
- `interface SPINNAKER_API_ABSTRACT` bool `Verify` = true) = 0

#### 8.79.1 Detailed Description

#### 8.79.2 Function Documentation

**8.79.2.1** `GenICam::gcstring GetValue ( bool Verify = false, bool IgnoreCache = false ) const` `[pure virtual]`

Get node value.

##### Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

##### Returns

The value read

**8.79.2.2** `GenlCam::gcstring operator() ( ) const` `[virtual]`

Get node value.

Execute the command.

**8.79.2.3** `virtual void Spinnaker::GenApi::operator= ( bool Value )` `[virtual]`

Set node value.

### 8.79.3 Variable Documentation

**8.79.3.1** `interface SPINNAKER_API_ABSTRACT IBoolean`

[Interface](#) for Boolean properties.

**8.79.3.2** `interface SPINNAKER_API_ABSTRACT bool Verify = true) = 0`



## 8.80 ICategory Interfaces

Collaboration diagram for ICategory Interfaces:



### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT ICategory](#)  
*Gives access to a category node.*

### 8.80.1 Detailed Description

### 8.80.2 Variable Documentation

#### 8.80.2.1 interface SPINNAKER\_API\_ABSTRACT ICategory

Gives access to a category node.

## 8.81 IChunkPort Interface

Collaboration diagram for IChunkPort Interface:



### Macros

- `#define` [CHUNK\\_BASE\\_ADDRESS\\_REGISTER](#) [GC\\_INT64\\_MAX](#)  
Address of a `int64_t` pseudo register containing the base address of the chunk (`MAX_INT64`)
- `#define` [CHUNK\\_BASE\\_ADDRESS\\_REGISTER\\_LEN](#) 8  
Length of the `CHUNK_BASE_ADDRESS_REGISTER` pseudo register.
- `#define` [CHUNK\\_LENGTH\\_REGISTER](#) ([GC\\_INT64\\_MAX](#)-15)  
Address of a `int64_t` pseudo register containing the length of the chunk.
- `#define` [CHUNK\\_LENGTH\\_REGISTER\\_LEN](#) 8  
Length of the `CHUNK_LENGTH_REGISTER` pseudo register.

### Functions

- virtual `EYesNo` [CacheChunkData](#) () const =0  
Indicates if the chunk a adapter must hold a cached version of the chunk data.

### Variables

- [interface](#) [SPINNAKER\\_API\\_ABSTRACT](#) [IChunkPort](#)  
*Interface* for ports attached to a chunk.

#### 8.81.1 Detailed Description

#### 8.81.2 Macro Definition Documentation

##### 8.81.2.1 `#define` [CHUNK\\_BASE\\_ADDRESS\\_REGISTER](#) [GC\\_INT64\\_MAX](#)

Address of a `int64_t` pseudo register containing the base address of the chunk (`MAX_INT64`)

##### 8.81.2.2 `#define` [CHUNK\\_BASE\\_ADDRESS\\_REGISTER\\_LEN](#) 8

Length of the `CHUNK_BASE_ADDRESS_REGISTER` pseudo register.

#### 8.81.2.3 `#define CHUNK_LENGTH_REGISTER (GC_INT64_MAX-15)`

Address of a `int64_t` pseudo register containing the length of the chunk.

#### 8.81.2.4 `#define CHUNK_LENGTH_REGISTER_LEN 8`

Length of the `CHUNK_LENGTH_REGISTER` pseudo register.

### 8.81.3 Function Documentation

#### 8.81.3.1 `virtual EYesNo Spinnaker::GenApi::CacheChunkData ( ) const` `[pure virtual]`

Indicates if the chunk a adapter must hold a cached version of the chunk data.

### 8.81.4 Variable Documentation

#### 8.81.4.1 `interface SPINNAKER_API_ABSTRACT IChunkPort`

[Interface](#) for ports attached to a chunk.

## 8.82 ICommand Interface

Collaboration diagram for ICommand Interface:



### Functions

- virtual bool [IsDone](#) (bool Verify=true)=0  
*Query whether the command is executed.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT ICommand](#)  
*Interface for command like properties.*

### 8.82.1 Detailed Description

### 8.82.2 Function Documentation

8.82.2.1 virtual bool Spinnaker::GenApi::IsDone ( bool *Verify* = true ) [pure virtual]

Query whether the command is executed.

#### Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
---------------	--

#### Returns

True if the Execute command has finished; false otherwise

### 8.82.3 Variable Documentation

8.82.3.1 [interface SPINNAKER\\_API\\_ABSTRACT ICommand](#)

[Interface](#) for command like properties.

## 8.83 IDestroy Interface

Collaboration diagram for IDestroy Interface:



### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IDestroy](#)  
*Interface to destroy an object.*

### 8.83.1 Detailed Description

### 8.83.2 Variable Documentation

#### 8.83.2.1 interface SPINNAKER\_API\_ABSTRACT IDestroy

##### Initial value:

```
{  
    virtual void Destroy() = 0  
}
```

[Interface](#) to destroy an object.

## 8.84 IDeviceInfo Interface

Collaboration diagram for IDeviceInfo Interface:



### Functions

- virtual `GenICam::gcstring GetVendorName ()=0`  
*Get the vendor name.*
- virtual `GenICam::gcstring GetToolTip ()=0`  
*Get tool tip.*
- virtual `GenICam::gcstring GetStandardNameSpace ()=0`  
*Get the standard name space.*
- virtual `void GetGenApiVersion (GenICam::Version_t &Version, uint16_t &Build)=0`  
*Get the version of the DLL's [GenApi](#) implementation.*
- virtual `void GetSchemaVersion (GenICam::Version_t &Version)=0`  
*Get the schema version number.*
- virtual `void GetDeviceVersion (GenICam::Version_t &Version)=0`  
*Get the version of the device description file.*
- virtual `GenICam::gcstring GetProductGuid ()=0`  
*Get the Guid describing the product.*
- virtual `GenICam::gcstring GetVersionGuid ()=0`  
*Get the Guid describing the product version.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IDeviceInfo](#)  
*Interface to get information about the device (= nodemap)*

#### 8.84.1 Detailed Description

#### 8.84.2 Function Documentation

8.84.2.1 `virtual void Spinnaker::GenApi::GetDeviceVersion ( GenICam::Version_t & Version ) [pure virtual]`

Get the version of the device description file.

8.84.2.2 `virtual void Spinnaker::GenApi::GetGenApiVersion ( GenICam::Version_t & Version, uint16_t & Build ) [pure virtual]`

Get the version of the DLL's [GenApi](#) implementation.

8.84.2.3 `virtual GenICam::gcstring Spinnaker::GenApi::GetProductGuid ( ) [pure virtual]`

Get the Guid describing the product.

8.84.2.4 `virtual void Spinnaker::GenApi::GetSchemaVersion ( GenICam::Version_t & Version ) [pure virtual]`

Get the schema version number.

8.84.2.5 `virtual GenICam::gcstring Spinnaker::GenApi::GetStandardNameSpace ( ) [pure virtual]`

Get the standard name space.

8.84.2.6 `GenICam::gcstring GetToolTip ( ) [pure virtual]`

Get tool tip.

Get a short description of the node.

8.84.2.7 `virtual GenICam::gcstring Spinnaker::GenApi::GetVendorName ( ) [pure virtual]`

Get the vendor name.

8.84.2.8 `virtual GenICam::gcstring Spinnaker::GenApi::GetVersionGuid ( ) [pure virtual]`

Get the Guid describing the product version.

### 8.84.3 Variable Documentation

8.84.3.1 `interface SPINNAKER_API_ABSTRACT IDeviceInfo`

**Initial value:**

```
{
    virtual GenICam::gcstring GetModelName() = 0
```

[Interface](#) to get information about the device (= nodemap)

## 8.85 IEnumEntry Interface

Collaboration diagram for IEnumEntry Interface:



### Functions

- virtual GenICam::gcstring [GetSymbolic](#) () const =0  
*Get symbolic enum value.*
- virtual double [GetNumericValue](#) ()=0  
*Get double number associated with the entry.*
- virtual bool [IsSelfClearing](#) ()=0  
*Indicates if the corresponding EnumEntry is self clearing.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IEnumEntry](#)  
*Interface of single enum value.*

#### 8.85.1 Detailed Description

#### 8.85.2 Function Documentation

**8.85.2.1** virtual double Spinnaker::GenApi::GetNumericValue ( ) [pure virtual]

Get double number associated with the entry.

**8.85.2.2** virtual GenICam::gcstring Spinnaker::GenApi::GetSymbolic ( ) const [pure virtual]

Get symbolic enum value.

**8.85.2.3** virtual bool Spinnaker::GenApi::IsSelfClearing ( ) [pure virtual]

Indicates if the corresponding EnumEntry is self clearing.

#### 8.85.3 Variable Documentation

**8.85.3.1** [interface SPINNAKER\\_API\\_ABSTRACT IEnumEntry](#)

[Interface](#) of single enum value.

Maps of Enum Values to symbolic values



## 8.86 IEnumeration Interface

Collaboration diagram for IEnumeration Interface:



### Functions

- virtual void [GetEntries](#) (NodeList\_t &Entries)=0  
*Get list of entry nodes.*
- virtual void [SetIntValue](#) (int64\_t Value, bool Verify=true)=0  
*Set integer node value.*
- virtual GenICam::gcstring [operator\\*](#) ()=0  
*Get string node value.*
- virtual int64\_t [GetIntValue](#) (bool Verify=false, bool IgnoreCache=false)=0  
*Get integer node value.*
- virtual IEnumEntry \* [GetEntryByName](#) (const GenICam::gcstring &Symbolic)=0  
*Get an entry node by name.*
- virtual IEnumEntry \* [GetEntry](#) (const int64\_t IntValue)=0  
*Get an entry node by its IntValue.*
- virtual IEnumEntry \* [GetCurrentEntry](#) (bool Verify=false, bool IgnoreCache=false)=0  
*Get the current entry.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IEnumeration](#)  
*Interface for enumeration properties.*

#### 8.86.1 Detailed Description

#### 8.86.2 Function Documentation

**8.86.2.1** IEnumEntry \* [GetCurrentEntry](#) ( bool *Verify* = false, bool *IgnoreCache* = false ) [pure virtual]

Get the current entry.

**8.86.2.2** virtual void Spinnaker::GenApi::GetEntries ( NodeList\_t & *Entries* ) [pure virtual]

Get list of entry nodes.

**8.86.2.3** `virtual IEnumEntry* Spinnaker::GenApi::GetEntry ( const int64_t IntValue ) [pure virtual]`

Get an entry node by its IntValue.

**8.86.2.4** `virtual IEnumEntry* Spinnaker::GenApi::GetEntryByName ( const GenICam::gcstring & Symbolic ) [pure virtual]`

Get an entry node by name.

**8.86.2.5** `virtual int64_t Spinnaker::GenApi::GetIntValue ( bool Verify = false, bool IgnoreCache = false ) [pure virtual]`

Get integer node value.

#### Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

#### Returns

The value read

**8.86.2.6** `GenICam::gcstring operator* ( ) [pure virtual]`

Get string node value.

Get node value.

**8.86.2.7** `virtual void Spinnaker::GenApi::SetIntValue ( int64_t Value, bool Verify = true ) [pure virtual]`

Set integer node value.

#### Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

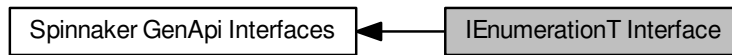
## 8.86.3 Variable Documentation

**8.86.3.1** `interface SPINNAKER_API_ABSTRACT IEnumeration`

[Interface](#) for enumeration properties.

## 8.87 IEnumerationT Interface

Collaboration diagram for IEnumerationT Interface:



### Functions

- virtual IEnumeration & **operator=** (EnumT Value)=0  
*Set node value.*
- virtual IEnumEntry \* **GetEntry** (const EnumT Value)=0  
*returns the EnumEntry object belonging to the Value*
- virtual IEnumeration & **operator=** (const GenICam::gcstring &ValueStr)=0  
*Set string node value.*

### Variables

- template<typename EnumT >  
**interface SPINNAKER\_API\_ABSTRACT IEnumerationT**  
*Interface for enumeration properties.*
- template<typename EnumT >  
**interface SPINNAKER\_API\_ABSTRACT virtual public IEnumReference**  
*Interface to construct an enum reference.*

#### 8.87.1 Detailed Description

#### 8.87.2 Function Documentation

**8.87.2.1** virtual IEnumEntry\* Spinnaker::GenApi::GetEntry ( const EnumT Value ) [pure virtual]

returns the EnumEntry object belonging to the Value

**8.87.2.2** virtual IEnumeration& Spinnaker::GenApi::operator= ( EnumT Value ) [pure virtual]

Set node value.

**8.87.2.3** `IString & operator= ( const GenICam::gcstring & ValueStr )` `[pure virtual]`

Set string node value.

Set node value.

Note : the operator= is not inherited thus the operator= versions from IEnumeration must be implemented again

### 8.87.3 Variable Documentation

**8.87.3.1** `interface SPINNAKER_API_ABSTRACT IEnumerationT`

[Interface](#) for enumeration properties.

**8.87.3.2** `interface SPINNAKER_API_ABSTRACT IEnumReference`

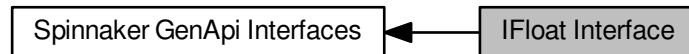
**Initial value:**

```
{  
    virtual void SetValue(EnumT Value, bool Verify = true) = 0
```

[Interface](#) to construct an enum reference.

## 8.88 IFloat Interface

Collaboration diagram for IFloat Interface:



### Functions

- virtual IFloat & [operator=](#) (double Value)=0  
*Set node value.*
- virtual double [GetMin](#) ()=0  
*Get minimum value allowed.*
- virtual double [GetMax](#) ()=0  
*Get maximum value allowed.*
- virtual bool [HasInc](#) ()=0  
*True if the float has a constant increment.*
- virtual EIncMode [GetIncMode](#) ()=0  
*Get increment mode.*
- virtual double [GetInc](#) ()=0  
*Get the constant increment if there is any.*
- virtual double \_autovector\_t [GetListOfValidValues](#) (bool bounded=true)=0  
*Get list of valid value.*
- virtual ERepresentation [GetRepresentation](#) ()=0  
*Get recommended representation.*
- virtual GenICam::gcstring [GetUnit](#) () const =0  
*Get the physical unit name.*
- virtual EDisplayNotation [GetDisplayNotation](#) () const =0  
*Get the way the float should be converted to a string.*
- virtual int64\_t [GetDisplayPrecision](#) () const =0  
*Get the precision to be used when converting the float to a string.*
- virtual void [ImposeMin](#) (double Value)=0  
*Restrict minimum value.*
- virtual void [ImposeMax](#) (double Value)=0  
*Restrict maximum value.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IFloat](#)  
*Interface for float properties.*

### 8.88.1 Detailed Description

### 8.88.2 Function Documentation

#### 8.88.2.1 `virtual EDisplayNotation Spinnaker::GenApi::GetDisplayNotation ( ) const` [pure virtual]

Get the way the float should be converted to a string.

#### 8.88.2.2 `virtual int64_t Spinnaker::GenApi::GetDisplayPrecision ( ) const` [pure virtual]

Get the precision to be used when converting the float to a string.

#### 8.88.2.3 `int64_t GetInc ( )` [pure virtual]

Get the constant increment if there is any.

Get increment.

#### 8.88.2.4 `EIncMode GetIncMode ( )` [pure virtual]

Get increment mode.

#### 8.88.2.5 `int64_autovector_t GetListOfValidValues ( bool bounded = true )` [pure virtual]

Get list of valid value.

#### 8.88.2.6 `int64_t GetMax ( )` [pure virtual]

Get maximum value allowed.

#### 8.88.2.7 `int64_t GetMin ( )` [pure virtual]

Get minimum value allowed.

#### 8.88.2.8 `ERepresentation GetRepresentation ( )` [pure virtual]

Get recommended representation.

#### 8.88.2.9 `GenICam::gcstring GetUnit ( ) const` [pure virtual]

Get the physical unit name.

8.88.2.10 virtual bool Spinnaker::GenApi::HasInc ( ) [pure virtual]

True if the float has a constant increment.

8.88.2.11 virtual void Spinnaker::GenApi::ImposeMax ( double *Value* ) [pure virtual]

Restrict maximum value.

8.88.2.12 virtual void Spinnaker::GenApi::ImposeMin ( double *Value* ) [pure virtual]

Restrict minimum value.

8.88.2.13 virtual IFloat& Spinnaker::GenApi::operator= ( double *Value* ) [pure virtual]

Set node value.

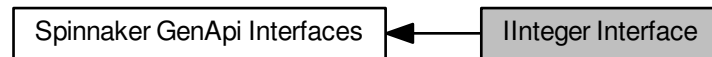
### 8.88.3 Variable Documentation

8.88.3.1 interface SPINNAKER\_API\_ABSTRACT IFloat

[Interface](#) for float properties.

## 8.89 Integer Interface

Collaboration diagram for Integer Interface:



### Functions

- virtual Integer & [operator=](#) (int64\_t Value)=0  
*Set node value.*
- virtual void [ImposeMin](#) (int64\_t Value)=0  
*Restrict minimum value.*
- virtual void [ImposeMax](#) (int64\_t Value)=0  
*Restrict maximum value.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT Integer](#)  
*Interface for integer properties.*

#### 8.89.1 Detailed Description

#### 8.89.2 Function Documentation

8.89.2.1 virtual void Spinnaker::GenApi::ImposeMax ( int64\_t Value ) [pure virtual]

Restrict maximum value.

8.89.2.2 virtual void Spinnaker::GenApi::ImposeMin ( int64\_t Value ) [pure virtual]

Restrict minimum value.

8.89.2.3 virtual Integer& Spinnaker::GenApi::operator= ( int64\_t Value ) [pure virtual]

Set node value.

#### 8.89.3 Variable Documentation

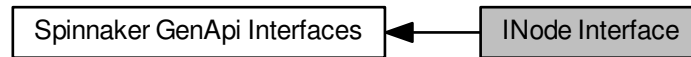
8.89.3.1 [interface SPINNAKER\\_API\\_ABSTRACT Integer](#)

[Interface](#) for integer properties.



## 8.90 INode Interface

Collaboration diagram for INode Interface:



### Functions

- virtual GenApi::ENamespace [GetNameSpace](#) () const =0  
*Get name space.*
- virtual EVisibility [GetVisibility](#) () const =0  
*Get the recommended visibility of the node.*
- virtual void [InvalidateNode](#) ()=0  
*Indicates that the node's value may have changed.*
- virtual bool [IsCacheable](#) () const =0  
*Is the node value cacheable.*
- virtual EYesNo [IsAccessModeCacheable](#) () const =0  
*True if the AccessMode can be cached.*
- virtual ECachingMode [GetCachingMode](#) () const =0  
*Get Caching Mode.*
- virtual int64\_t [GetPollingTime](#) () const =0  
*recommended polling time (for non-cacheable nodes)*
- virtual GenICam::gcstring [GetDescription](#) () const =0  
*Get a long description of the node.*
- virtual GenICam::gcstring [GetDisplayName](#) () const =0  
*Get a name string for display.*
- virtual void [GetChildren](#) (GenApi::NodeList\_t &Children, ELinkType LinkType=ctReadingChildren) const =0  
*Get all nodes this node directly depends on.*
- virtual void [GetParents](#) (GenApi::NodeList\_t &Parents) const =0  
*Gets all nodes this node is directly depending on.*
- virtual CallbackHandleType [RegisterCallback](#) (CNodeCallback \*pCallback)=0  
*Register change callback Takes ownership of the CNodeCallback object.*
- virtual bool [DeregisterCallback](#) (CallbackHandleType hCallback)=0  
*De register change callback Destroys CNodeCallback object.*
- virtual INodeMap \* [GetNodeMap](#) () const =0  
*Retrieves the central node map.*
- virtual GenICam::gcstring [GetEventID](#) () const =0  
*Get the EventId of the node.*
- virtual bool [IsStreamable](#) () const =0  
*True if the node is streamable.*
- virtual void [GetPropertyNames](#) (GenICam::gcstring\_vector &PropertyNames) const =0  
*Returns a list of the names all properties set during initialization.*

- virtual bool [GetProperty](#) (const GenlCam::gcstring &PropertyName, GenlCam::gcstring &ValueStr, GenlCam::gcstring &AttributeStr)=0  
*Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.*
- virtual void [ImposeAccessMode](#) (EAccessMode ImposedAccessMode)=0  
*Imposes an access mode to the natural access mode of the node.*
- virtual void [ImposeVisibility](#) (EVisibility ImposedVisibility)=0  
*Imposes a visibility to the natural visibility of the node.*
- virtual INode \* [GetAlias](#) () const =0  
*Retrieves the a node which describes the same feature in a different way.*
- virtual INode \* [GetCastAlias](#) () const =0  
*Retrieves the a node which describes the same feature so that it can be casted.*
- virtual GenlCam::gcstring [GetDocuURL](#) () const =0  
*Gets a URL pointing to the documentation of that feature.*
- virtual bool [IsDeprecated](#) () const =0  
*True if the node should not be used any more.*
- virtual EInterfaceType [GetPrincipalInterfaceType](#) () const =0  
*Get the type of the main interface of a node.*
- virtual bool [IsFeature](#) () const =0  
*True if the node can be reached via category nodes from a category node named "Root".*
- virtual bool [operator==](#) (int nullPtr) const =0
- virtual bool [operator!=](#) (int nullPtr) const =0
- bool [IsReadable](#) (EAccessMode AccessMode)  
*Tests if readable.*
- bool [IsReadable](#) (const IBase \*p)  
*Checks if a node is readable.*
- bool [IsReadable](#) (const IBase &r)  
*Checks if a node is readable.*
- bool [IsWritable](#) (EAccessMode AccessMode)  
*Tests if writable.*
- bool [IsWritable](#) (const IBase \*p)  
*Checks if a node is writable.*
- bool [IsWritable](#) (const IBase &r)  
*Checks if a node is writable.*
- bool [IsImplemented](#) (EAccessMode AccessMode)  
*Tests if implemented.*
- bool [IsImplemented](#) (const IBase \*p)  
*Checks if a node is implemented.*
- bool [IsImplemented](#) (const IBase &r)  
*Checks if a node is implemented.*
- bool [IsAvailable](#) (EAccessMode AccessMode)  
*Tests if available.*
- bool [IsAvailable](#) (const IBase \*p)  
*Checks if a node is available.*
- bool [IsAvailable](#) (const IBase &r)  
*Checks if a node is available.*
- EAccessMode [Combine](#) (EAccessMode Peter, EAccessMode Paul)  
*Computes which access mode the two guards allow together.*
- bool [IsVisible](#) (EVisibility Visibility, EVisibility MaxVisibility)  
*Tests Visibility CAVE : this relies on the EVisibility enum's coding.*
- EVisibility [Combine](#) (EVisibility Peter, EVisibility Paul)

- *Computes which visibility the two guards allow together.*
- bool [IsCacheable](#) (ECachingMode CachingMode)  
*Tests Cacheability.*
- ECachingMode [Combine](#) (ECachingMode Peter, ECachingMode Paul)  
*Computes which CachingMode results from a combination.*

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT INode](#)  
*Interface common to all nodes.*
- [interface SPINNAKER\\_API\\_ABSTRACT](#) virtual public [IReference](#)  
*Interface to construct a reference.*

## 8.90.1 Detailed Description

## 8.90.2 Function Documentation

8.90.2.1 **EAccessMode Spinnaker::GenApi::Combine ( EAccessMode *Peter*, EAccessMode *Paul* )** `[inline]`

Computes which access mode the two guards allow together.

8.90.2.2 **EVisibility Spinnaker::GenApi::Combine ( EVisibility *Peter*, EVisibility *Paul* )** `[inline]`

Computes which visibility the two guards allow together.

8.90.2.3 **ECachingMode Spinnaker::GenApi::Combine ( ECachingMode *Peter*, ECachingMode *Paul* )** `[inline]`

Computes which CachingMode results from a combination.

8.90.2.4 **virtual bool Spinnaker::GenApi::DeregisterCallback ( CallbackHandleType *hCallback* )** `[pure virtual]`

De register change callback Destroys [CNodeCallback](#) object.

### Returns

true if the callback handle was valid

8.90.2.5 **virtual INode\* Spinnaker::GenApi::GetAlias ( ) const** `[pure virtual]`

Retrieves the a node which describes the same feature in a different way.

8.90.2.6 **virtual ECachingMode Spinnaker::GenApi::GetCachingMode ( ) const** `[pure virtual]`

Get Caching Mode.

8.90.2.7 **virtual INode\* Spinnaker::GenApi::GetCastAlias ( ) const** `[pure virtual]`

Retrieves the a node which describes the same feature so that it can be casted.

8.90.2.8 **virtual void Spinnaker::GenApi::GetChildren ( GenApi::NodeList\_t & *Children*, ELinkType *LinkType* = ctReadingChildren ) const** `[pure virtual]`

Get all nodes this node directly depends on.

## Parameters

out	<i>Children</i>	List of children nodes
	<i>LinkType</i>	The link type

8.90.2.9 `virtual GenlCam::gcstring Spinnaker::GenApi::GetDescription ( ) const` [pure virtual]

Get a long description of the node.

8.90.2.10 `virtual GenlCam::gcstring Spinnaker::GenApi::GetDisplayName ( ) const` [pure virtual]

Get a name string for display.

8.90.2.11 `virtual GenlCam::gcstring Spinnaker::GenApi::GetDocuURL ( ) const` [pure virtual]

Gets a URL pointing to the documentation of that feature.

8.90.2.12 `virtual GenlCam::gcstring Spinnaker::GenApi::GetEventID ( ) const` [pure virtual]

Get the EventId of the node.

8.90.2.13 `virtual GenApi::ENamespace Spinnaker::GenApi::GetNameSpace ( ) const` [pure virtual]

Get name space.

8.90.2.14 `virtual INodeMap* Spinnaker::GenApi::GetNodeMap ( ) const` [pure virtual]

Retrieves the central node map.

8.90.2.15 `virtual void Spinnaker::GenApi::GetParents ( GenApi::NodeList_t & Parents ) const` [pure virtual]

Gets all nodes this node is directly depending on.

## Parameters

out	<i>Parents</i>	List of parent nodes
-----	----------------	----------------------

8.90.2.16 `virtual int64_t Spinnaker::GenApi::GetPollingTime ( ) const` [pure virtual]

recommended polling time (for non-cacheable nodes)

8.90.2.17 `virtual EInterfaceType Spinnaker::GenApi::GetPrincipalInterfaceType ( ) const [pure virtual]`

Get the type of the main interface of a node.

8.90.2.18 `virtual bool Spinnaker::GenApi::GetProperty ( const GenICam::gcstring & PropertyName, GenICam::gcstring & ValueStr, GenICam::gcstring & AttributeStr ) [pure virtual]`

Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.

8.90.2.19 `virtual void Spinnaker::GenApi::GetPropertyNames ( GenICam::gcstring_vector & PropertyNames ) const [pure virtual]`

Returns a list of the names all properties set during initialization.

8.90.2.20 `virtual EVisibility Spinnaker::GenApi::GetVisibility ( ) const [pure virtual]`

Get the recommended visibility of the node.

8.90.2.21 `virtual void Spinnaker::GenApi::ImposeAccessMode ( EAccessMode ImposedAccessMode ) [pure virtual]`

Imposes an access mode to the natural access mode of the node.

8.90.2.22 `virtual void Spinnaker::GenApi::ImposeVisibility ( EVisibility ImposedVisibility ) [pure virtual]`

Imposes a visibility to the natural visibility of the node.

8.90.2.23 `virtual void Spinnaker::GenApi::InvalidateNode ( ) [pure virtual]`

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

8.90.2.24 `virtual EYesNo Spinnaker::GenApi::IsAccessModeCacheable ( ) const [pure virtual]`

True if the AccessMode can be cached.

8.90.2.25 `bool Spinnaker::GenApi::IsAvailable ( EAccessMode AccessMode ) [inline]`

Tests if available.

8.90.2.26 `bool Spinnaker::GenApi::IsAvailable ( const IBase * p ) [inline]`

Checks if a node is available.

8.90.2.27 `bool Spinnaker::GenApi::IsAvailable ( const IBase & r ) [inline]`

Checks if a node is available.

8.90.2.28 `virtual bool Spinnaker::GenApi::IsCachable ( ) const [pure virtual]`

Is the node value cacheable.

8.90.2.29 `bool Spinnaker::GenApi::IsCacheable ( ECacheMode CachingMode ) [inline]`

Tests Cacheability.

8.90.2.30 `virtual bool Spinnaker::GenApi::IsDeprecated ( ) const [pure virtual]`

True if the node should not be used any more.

8.90.2.31 `virtual bool Spinnaker::GenApi::IsFeature ( ) const [pure virtual]`

True if the node can be reached via category nodes from a category node named "Root".

8.90.2.32 `bool Spinnaker::GenApi::IsImplemented ( EAccessMode AccessMode ) [inline]`

Tests if implemented.

8.90.2.33 `bool Spinnaker::GenApi::IsImplemented ( const IBase * p ) [inline]`

Checks if a node is implemented.

8.90.2.34 `bool Spinnaker::GenApi::IsImplemented ( const IBase & r ) [inline]`

Checks if a node is implemented.

8.90.2.35 `bool Spinnaker::GenApi::IsReadable ( EAccessMode AccessMode ) [inline]`

Tests if readable.

8.90.2.36 `bool Spinnaker::GenApi::IsReadable ( const IBase * p ) [inline]`

Checks if a node is readable.

8.90.2.37 `bool Spinnaker::GenApi::IsReadable ( const IBase & r ) [inline]`

Checks if a node is readable.

8.90.2.38 `virtual bool Spinnaker::GenApi::IsStreamable ( ) const [pure virtual]`

True if the node is streamable.

8.90.2.39 `bool Spinnaker::GenApi::IsVisible ( EVisibility Visibility, EVisibility MaxVisibility ) [inline]`

Tests Visibility CAVE : this relies on the EVisibility enum's coding.

8.90.2.40 `bool Spinnaker::GenApi::IsWritable ( EAccessMode AccessMode ) [inline]`

Tests if writable.

8.90.2.41 `bool Spinnaker::GenApi::IsWritable ( const IBase * p ) [inline]`

Checks if a node is writable.

8.90.2.42 `bool Spinnaker::GenApi::IsWritable ( const IBase & r ) [inline]`

Checks if a node is writable.

8.90.2.43 `virtual bool Spinnaker::GenApi::operator!= ( int nullptr ) const [pure virtual]`

8.90.2.44 `virtual bool Spinnaker::GenApi::operator== ( int nullptr ) const [pure virtual]`

8.90.2.45 `virtual CallbackHandleType Spinnaker::GenApi::RegisterCallback ( CNodeCallback * pCallback ) [pure virtual]`

Register change callback Takes ownership of the [CNodeCallback](#) object.

## 8.90.3 Variable Documentation

8.90.3.1 `interface SPINNAKER_API_ABSTRACT INode`

[Interface](#) common to all nodes.

8.90.3.2 `interface SPINNAKER_API_ABSTRACT IReference`

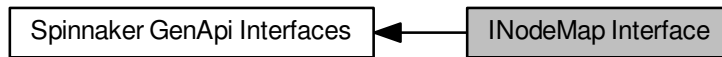
**Initial value:**

```
{
    virtual GenICam::gcstring GetName(bool FullQualified = false) const = 0
```

[Interface](#) to construct a reference.

## 8.91 INodeMap Interface

Collaboration diagram for INodeMap Interface:



### Functions

- virtual `INode *` [GetNode](#) (const `GenICam::gcstring &Name`) const =0  
*Retrieves the node from the central map by Name.*
- virtual void [InvalidateNodes](#) () const =0  
*Invalidates all nodes.*
- virtual bool [Connect](#) (`IPort *pPort`, const `GenICam::gcstring &PortName`) const =0  
*Connects a port to a port node with given name.*
- virtual bool [Connect](#) (`IPort *pPort`) const =0  
*Connects a port to the standard port "Device".*
- virtual void [Poll](#) (`int64_t ElapsedTime`)=0  
*Fires nodes which have a polling time.*
- virtual `CLock &` [GetLock](#) () const =0  
*Returns the lock which guards the node map.*
- virtual `uint64_t` [GetNumNodes](#) () const =0  
*Get the number of nodes in the map.*
- virtual `GenICam::gcstring` [GetDeviceName](#) () const =0  
*Get a name of the device.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT INodeMap](#)  
*Interface to access the node map.*

#### 8.91.1 Detailed Description

#### 8.91.2 Function Documentation

- 8.91.2.1 `virtual bool Spinnaker::GenApi::Connect ( IPort * pPort, const GenICam::gcstring & PortName ) const` [pure virtual]

Connects a port to a port node with given name.



8.91.2.2 `virtual bool Spinnaker::GenApi::Connect ( IPort * pPort ) const` [pure virtual]

Connects a port to the standard port "Device".

8.91.2.3 `GenICam::gcstring GetDeviceName ( )` [pure virtual]

Get a name of the device.

Get device name The device name identifies a device instance, e.g.

for debugging purposes. The default is "Device".

8.91.2.4 `virtual CLock& Spinnaker::GenApi::GetLock ( ) const` [pure virtual]

Returns the lock which guards the node map.

8.91.2.5 `virtual INode* Spinnaker::GenApi::GetNode ( const GenICam::gcstring & Name ) const` [pure virtual]

Retrieves the node from the central map by Name.

8.91.2.6 `virtual uint64_t Spinnaker::GenApi::GetNumNodes ( ) const` [pure virtual]

Get the number of nodes in the map.

8.91.2.7 `virtual void Spinnaker::GenApi::InvalidateNodes ( ) const` [pure virtual]

Invalidates all nodes.

8.91.2.8 `virtual void Spinnaker::GenApi::Poll ( int64_t ElapsedTime )` [pure virtual]

Fires nodes which have a polling time.

### 8.91.3 Variable Documentation

8.91.3.1 `interface SPINNAKER_API_ABSTRACT INodeMap`

**Initial value:**

```
{
    virtual void GetNodes(NodeList_t &Nodes) const = 0
}
```

Interface to access the node map.

## 8.92 INodeMapDyn Interface

Collaboration diagram for INodeMapDyn Interface:



### Functions

- virtual void [LoadXMLFromFile](#) (const GenICam::gcstring &FileName)=0  
*Loads an XML from a file.*
- virtual void [LoadXMLFromFileInject](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)=0  
*Loads an XML from a file with injection.*
- virtual void [LoadXMLFromString](#) (const GenICam::gcstring &XMLData)=0  
*Loads an XML from a string.*
- virtual void [LoadXMLFromStringInject](#) (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)=0  
*Loads an XML from a string with injection.*
- virtual void [PreprocessXMLFromFile](#) (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32\_t XMLValidation=xv↵Default)=0  
*Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.*
- virtual void [MergeXMLFiles](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectedFileName, const GenICam::gcstring &OutputFileName)=0  
*Injects an XML file into a target file.*
- virtual void [ExtractIndependentSubtree](#) (const GenICam::gcstring &XMLData, const GenICam::gcstring &SubTreeRootNodeName, GenICam::gcstring &Extracted↵Subtree)=0  
*Extract independent subtree.*
- virtual void [GetSupportedSchemaVersions](#) (GenICam::gcstring\_vector &SchemaVersions)=0  
*Gets a list of supported schema versions.*
- virtual void [LoadXMLFromZIPFile](#) (const GenICam::gcstring &ZipFileName)=0  
*Loads an XML from a ZIP file.*
- virtual void [LoadXMLFromZIPData](#) (const void \*zipData, size\_t zipSize)=0  
*Loads an XML from a ZIP data buffer.*
- virtual void [PreprocessXMLFromZIPFile](#) (const GenICam::gcstring &XMLFileName, const GenICam::gcstring &StyleSheetFileName, const GenICam::gcstring &OutputFileName, const uint32\_t XMLValidation=xv↵Default)=0  
*Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT INodeMapDyn](#)  
*Interface to access the node map.*

### 8.92.1 Detailed Description

### 8.92.2 Function Documentation

**8.92.2.1** `virtual void Spinnaker::GenApi::ExtractIndependentSubtree ( const GenICam::gcstring & XMLData,  
const GenICam::gcstring & InjectXMLData, const GenICam::gcstring & SubTreeRootNodeName,  
GenICam::gcstring & ExtractedSubtree ) [pure virtual]`

Extract independent subtree.

#### Parameters

<i>InjectXMLData</i>	The XML data the subtree is extracted from.
<i>SubTreeRootNodeName</i>	Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed.
<i>ExtractedSubtree</i>	The name of the node that represents the root of the subtree that shall be extracted.> The returned extracted subtree as string.

**8.92.2.2** `virtual void Spinnaker::GenApi::GetSupportedSchemaVersions ( GenICam::gcstring_vector & SchemaVersions )  
[pure virtual]`

Gets a list of supported schema versions.

Each list entry is a string with the format "<Major>.<Minor>" where <Major> and <Minor> are integers Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

**8.92.2.3** `virtual void Spinnaker::GenApi::LoadXMLFromFile ( const GenICam::gcstring & FileName ) [pure  
virtual]`

Loads an XML from a file.

**8.92.2.4** `virtual void Spinnaker::GenApi::LoadXMLFromFileInject ( const GenICam::gcstring & TargetFileName, const  
GenICam::gcstring & InjectFileName ) [pure virtual]`

Loads an XML from a file with injection.

**8.92.2.5** `virtual void Spinnaker::GenApi::LoadXMLFromString ( const GenICam::gcstring & XMLData ) [pure  
virtual]`

Loads an XML from a string.

**8.92.2.6** `virtual void Spinnaker::GenApi::LoadXMLFromStringInject ( const GenICam::gcstring & TargetXMLData, const GenICam::gcstring & InjectXMLData ) [pure virtual]`

Loads an XML from a string with injection.

**8.92.2.7** `virtual void Spinnaker::GenApi::LoadXMLFromZIPData ( const void * zipData, size_t zipSize ) [pure virtual]`

Loads an XML from a ZIP data buffer.

**8.92.2.8** `virtual void Spinnaker::GenApi::LoadXMLFromZIPFile ( const GenICam::gcstring & ZipFileName ) [pure virtual]`

Loads an XML from a ZIP file.

**8.92.2.9** `virtual void Spinnaker::GenApi::MergeXMLFiles ( const GenICam::gcstring & TargetFileName, const GenICam::gcstring & InjectedFileName, const GenICam::gcstring & OutputFileName ) [pure virtual]`

Injects an XML file into a target file.

#### Parameters

<i>InjectedFileName</i>	Name of the target XML file to process
<i>OutputFileName</i>	Name of the Injected XML file to process> Name of the output file

**8.92.2.10** `virtual void Spinnaker::GenApi::PreprocessXMLFromFile ( const GenICam::gcstring & XMLFileName, const GenICam::gcstring & StyleSheetFileName, const GenICam::gcstring & OutputFileName, const uint32_t XMLValidation = xvDefault ) [pure virtual]`

Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

#### Parameters

<i>StyleSheetFileName</i>	The name of the XML file to process
<i>OutputFileName</i>	Optional name of a style sheet which is applied after the pre-processor (can be empty string)> This has no effect if the <i>OutputFileName</i> is an empty string

## Parameters

<i>XML Validation</i>	Optional name of an output file into which the processed data is written (can be empty string)> Optional bit mask formed from EXMLValidation enumeration indicating which tests should be performed on the XML file
-----------------------	---

8.92.2.11 `virtual void Spinnaker::GenApi::PreprocessXMLFromZIPFile ( const GenICam::gcstring & XMLFileName, const GenICam::gcstring & StyleSheetFileName, const GenICam::gcstring & OutputFileName, const uint32_t XMLValidation = xvDefault ) [pure virtual]`

Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

## Parameters

<i>StyleSheetFileName</i>	The name of the XML file to process
<i>OutputFileName</i>	Optional name of a style sheet which is applied after the pre-processor (can be empty string)> This has no effect if the OutputFileName is an empty string
<i>XML Validation</i>	Optional name of an output file into which the processed data is written (can be empty string)> Optional bit mask formed from EXMLValidation enumeration indicating which tests should be performed on the XML file

## 8.92.3 Variable Documentation

8.92.3.1 `interface SPINNAKER_API_ABSTRACT INodeMapDyn`

[Interface](#) to access the node map.

## 8.93 IntegerNode Class

Collaboration diagram for IntegerNode Class:



### Classes

- class [IntegerNode](#)  
*[Interface](#) for string properties.*

### Typedefs

- typedef IntegerNode [CIntegerRef](#)

#### 8.93.1 Detailed Description

#### 8.93.2 Typedef Documentation

##### 8.93.2.1 typedef IntegerNode CIntegerRef

## 8.94 IntRegNode Class

Collaboration diagram for IntRegNode Class:



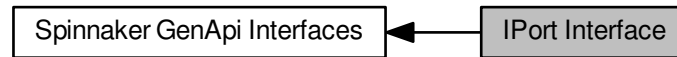
### Classes

- class [IntRegNode](#)  
*Interface for string properties.*

#### 8.94.1 Detailed Description

## 8.95 IPort Interface

Collaboration diagram for IPort Interface:



### Functions

- virtual void [Write](#) (const void \*pBuffer, int64\_t Address, int64\_t Length)=0  
*Writes a chunk of bytes to the port.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IPort](#)  
*Interface for ports.*
- [interface SPINNAKER\\_API\\_ABSTRACT int64\\_t Address](#)
- [interface SPINNAKER\\_API\\_ABSTRACT int64\\_t int64\\_t Length = 0](#)

#### 8.95.1 Detailed Description

#### 8.95.2 Function Documentation

8.95.2.1 virtual void Spinnaker::GenApi::Write ( const void \* *pBuffer*, int64\_t *Address*, int64\_t *Length* ) [pure virtual]

Writes a chunk of bytes to the port.

#### 8.95.3 Variable Documentation

8.95.3.1 interface SPINNAKER\_API\_ABSTRACT int64\_t Address

8.95.3.2 interface SPINNAKER\_API\_ABSTRACT IPort

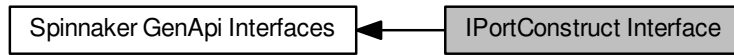
[Interface](#) for ports.

8.95.3.3 interface SPINNAKER\_API\_ABSTRACT int64\_t Length = 0



## 8.96 IPortConstruct Interface

Collaboration diagram for IPortConstruct Interface:



### Functions

- virtual EYesNo [GetSwapEndianness](#) ()=0  
*Determines if the port adapter must perform an endianness swap.*

### Variables

- [interface SPINNAKER\\_API IPortConstruct](#)  
*Interface for ports.*

### 8.96.1 Detailed Description

### 8.96.2 Function Documentation

8.96.2.1 virtual EYesNo Spinnaker::GenApi::GetSwapEndianness ( ) [pure virtual]

Determines if the port adapter must perform an endianness swap.

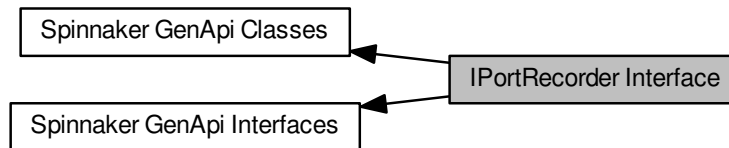
### 8.96.3 Variable Documentation

8.96.3.1 [interface SPINNAKER\\_API IPortConstruct](#)

[Interface](#) for ports.

## 8.97 IPortRecorder Interface

Collaboration diagram for IPortRecorder Interface:



### Functions

- virtual void [Replay](#) (IPort \*pPort)=0  
*Replays the write command to the given port interface.*
- virtual void [SetCookie](#) (const int64\_t Value)=0  
*Sets a cookie in case the port implementation want to cache a command list.*
- virtual int64\_t [GetCookie](#) ()=0  
*Gets the cookie a port implementation may have set for caching a command list.*
- virtual void [StopRecording](#) ()=0  
*Stops recording.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IPortWriteList](#)
- [interface SPINNAKER\\_API\\_ABSTRACT IPortReplay](#)  
*Interface for replaying write commands on a port.*
- [interface SPINNAKER\\_API\\_ABSTRACT bool Invalidate = true\) = 0](#)
- [interface SPINNAKER\\_API\\_ABSTRACT IPortRecorder](#)  
*Interface for recording write commands on a port.*

### 8.97.1 Detailed Description

### 8.97.2 Function Documentation

**8.97.2.1** virtual int64\_t Spinnaker::GenApi::GetCookie ( ) [pure virtual]

Gets the cookie a port implementation may have set for caching a command list.

**8.97.2.2** virtual void Spinnaker::GenApi::Replay ( IPort \* pPort ) [pure virtual]

Replays the write command to the given port interface.

8.97.2.3 `virtual void Spinnaker::GenApi::SetCookie ( const int64_t Value ) [pure virtual]`

Sets a cookie in case the port implementation want to cache a command list.

8.97.2.4 `virtual void Spinnaker::GenApi::StopRecording ( ) [pure virtual]`

Stops recording.

### 8.97.3 Variable Documentation

8.97.3.1 `interface SPINNAKER_API_ABSTRACT bool Invalidate = true) = 0`

8.97.3.2 `interface SPINNAKER_API_ABSTRACT IPortRecorder`

[Interface](#) for recording write commands on a port.

8.97.3.3 `interface SPINNAKER_API_ABSTRACT IPortReplay`

[Interface](#) for replaying write commands on a port.

8.97.3.4 `interface SPINNAKER_API_ABSTRACT IPortWriteList`

**Initial value:**

```
{  
    virtual void Write(const void *pBuffer, int64_t Address, int64_t  
    Length) = 0
```

## 8.98 IRegister Interfaces

Collaboration diagram for IRegister Interfaces:



### Functions

- virtual void [Get](#) (uint8\_t \*pBuffer, int64\_t Length, bool Verify=false, bool IgnoreCache=false)=0  
*Fills a buffer with the register's contents.*
- virtual int64\_t [GetLength](#) ()=0  
*Retrieves the Length of the register [Bytes].*
- virtual int64\_t [GetAddress](#) ()=0  
*Retrieves the Address of the register.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IRegister](#)  
*Interface for registers.*

### 8.98.1 Detailed Description

### 8.98.2 Function Documentation

**8.98.2.1** virtual void Spinnaker::GenApi::Get ( uint8\_t \* *pBuffer*, int64\_t *Length*, bool *Verify* = false, bool *IgnoreCache* = false ) [pure virtual]

Fills a buffer with the register's contents.

#### Parameters

<i>pBuffer</i>	The buffer receiving the data to read
<i>Length</i>	The number of bytes to retrieve
<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

#### Returns

The value read

8.98.2.2 `virtual int64_t Spinnaker::GenApi::GetAddress ( ) [pure virtual]`

Retrieves the Address of the register.

8.98.2.3 `virtual int64_t Spinnaker::GenApi::GetLength ( ) [pure virtual]`

Retrieves the Length of the register [Bytes].

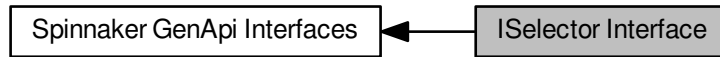
### 8.98.3 Variable Documentation

8.98.3.1 `interface SPINNAKER_API_ABSTRACT IRegister`

[Interface](#) for registers.

## 8.99 ISelector Interface

Collaboration diagram for ISelector Interface:



### Functions

- virtual void [GetSelectedFeatures](#) (FeatureList\_t &) const =0  
*retrieve the group of selected features*
- virtual void [GetSelectingFeatures](#) (FeatureList\_t &) const =0  
*retrieve the group of features selecting this node*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT ISelector](#)  
*Interface for groups of features selected by a single one.*

#### 8.99.1 Detailed Description

#### 8.99.2 Function Documentation

8.99.2.1 virtual void Spinnaker::GenApi::GetSelectedFeatures ( FeatureList\_t & ) const [pure virtual]

retrieve the group of selected features

8.99.2.2 virtual void Spinnaker::GenApi::GetSelectingFeatures ( FeatureList\_t & ) const [pure virtual]

retrieve the group of features selecting this node

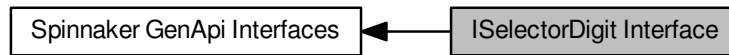
#### 8.99.3 Variable Documentation

8.99.3.1 interface SPINNAKER\_API\_ABSTRACT ISelector

[Interface](#) for groups of features selected by a single one.

## 8.100 ISelectorDigit Interface

Collaboration diagram for ISelectorDigit Interface:



### Functions

- virtual bool [SetNext](#) (bool Tick=true)=0  
*Sets digit to next value.*
- virtual void [Restore](#) ()=0  
*Restores the selectors' values found at creation.*
- virtual GenICam::gcstring [ToString](#) ()=0  
*Returns a string representation of the digit.*
- virtual void [GetSelectorList](#) (FeatureList\_t &SelectorList, bool Incremental=false)=0  
*Retrieves an ordered list of selectors.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT ISelectorDigit](#)  
*Interface of a "digit" of the "counter" formed by the selector set.*

#### 8.100.1 Detailed Description

#### 8.100.2 Function Documentation

8.100.2.1 `virtual void Spinnaker::GenApi::GetSelectorList ( FeatureList_t & SelectorList, bool Incremental = false )`  
[pure virtual]

Retrieves an ordered list of selectors.

##### Parameters

<i>Incremental</i>	List to contain the selector pointer> if true only seletor changed since the last GetNext are contained
--------------------	---

8.100.2.2 `virtual void Spinnaker::GenApi::Restore ( ) [pure virtual]`

Restores the selectors' values found at creation.

8.100.2.3 `virtual bool Spinnaker::GenApi::SetNext ( bool Tick =true ) [pure virtual]`

Sets digit to next value.

#### Parameters

<i>Tick</i>	if false the counter does not tick (but realize it could have)
-------------	--

#### Returns

true if the resulting value is valid

8.100.2.4 `virtual GenICam::gcstring Spinnaker::GenApi::ToString ( ) [pure virtual]`

Returns a string representation of the digit.

### 8.100.3 Variable Documentation

8.100.3.1 `interface SPINNAKER_API_ABSTRACT ISelectorDigit`

#### Initial value:

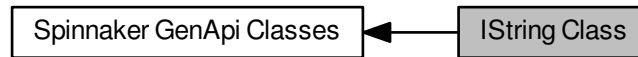
```
{
    virtual bool SetFirst() = 0
```

[Interface](#) of a "digit" of the "counter" formed by the selector set.



## 8.101 IString Class

Collaboration diagram for IString Class:



### Functions

- virtual int64\_t [GetMaxLength](#) ()=0  
*Retrieves the maximum length of the string in bytes.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IString](#)  
*Interface for string properties.*

#### 8.101.1 Detailed Description

#### 8.101.2 Function Documentation

8.101.2.1 virtual int64\_t Spinnaker::GenApi::GetMaxLength ( ) [pure virtual]

Retrieves the maximum length of the string in bytes.

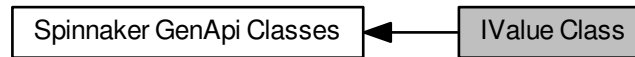
#### 8.101.3 Variable Documentation

8.101.3.1 interface SPINNAKER\_API\_ABSTRACT IString

[Interface](#) for string properties.

## 8.102 IValue Class

Collaboration diagram for IValue Class:



### Functions

- virtual GenICam::gcstring [ToString](#) (bool Verify=false, bool IgnoreCache=false)=0  
*Get content of the node as string.*
- virtual void [FromString](#) (const GenICam::gcstring &ValueStr, bool Verify=true)=0  
*Set content of the node as string.*
- virtual bool [IsValueCacheValid](#) () const =0  
*Checks if the value comes from cache or is requested from another node.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IValue](#)  
*Interface for value properties.*

### 8.102.1 Detailed Description

### 8.102.2 Function Documentation

**8.102.2.1** virtual void Spinnaker::GenApi::FromString ( const GenICam::gcstring & ValueStr, bool Verify = true )  
[pure virtual]

Set content of the node as string.

#### Parameters

<i>ValueStr</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

**8.102.2.2** virtual bool Spinnaker::GenApi::IsValueCacheValid ( ) const [pure virtual]

Checks if the value comes from cache or is requested from another node.

8.102.2.3 `virtual GenICam::gcstring Spinnaker::GenApi::ToString ( bool Verify = false, bool IgnoreCache = false )`  
[pure virtual]

Get content of the node as string.

#### Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

#### Returns

The value read

### 8.102.3 Variable Documentation

8.102.3.1 `interface SPINNAKER_API_ABSTRACT IValue`

[Interface](#) for value properties.

## 8.103 Node Class

Collaboration diagram for Node Class:



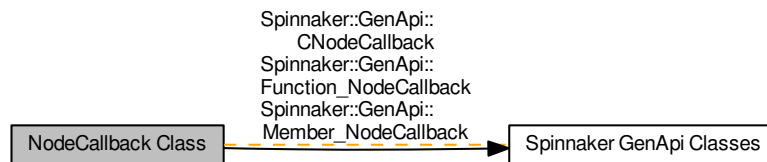
### Classes

- class [Node](#)  
*class common to all nodes*

#### 8.103.1 Detailed Description

## 8.104 NodeCallback Class

Collaboration diagram for NodeCallback Class:



### Classes

- class [CNodeCallback](#)  
*callback body instance for INode pointers*
- class [Function\\_NodeCallback](#)< [Function](#) >  
*Container for a function pointer.*
- class [Member\\_NodeCallback](#)< [Client](#), [Member](#) >  
*Container for a member function pointer.*

### Enumerations

- enum [ECallbackType](#) {  
  [cbPostInsideLock](#) = 1,  
  [cbPostOutsideLock](#) = 2 }  
*the type of callback*

### Functions

- template<class [Function](#) >  
  [CNodeCallback](#) \* [make\\_NodeCallback](#) ([INode](#) \*pNode, [Function](#) function, [ECallbackType](#) CallbackType)  
    *make a new callback object for C functions*
- template<class [Function](#) >  
  intptr\_t [Register](#) ([INode](#) \*pNode, [Function](#) f, [ECallbackType](#) CallbackType=cbPostInsideLock)  
    *Register a C-function as a callback.*
- template<class [Client](#) , class [Member](#) >  
  [CNodeCallback](#) \* [make\\_NodeCallback](#) ([INode](#) \*pNode, [Client](#) &client, [Member](#) member, [ECallbackType](#) CallbackType)  
    *make a new callback object for member functions*
- template<class [Client](#) , class [Member](#) >  
  intptr\_t [Register](#) ([INode](#) \*pNode, [Client](#) &c, [Member](#) m, [ECallbackType](#) CallbackType=cbPostInsideLock)  
    *Register a C++-member function a callback.*
- [SPINNAKER\\_API](#) void [Deregister](#) ([GenApi::CallbackHandleType](#) pCallbackInfo)  
    *Unregistering callback by handle.*

### 8.104.1 Detailed Description

### 8.104.2 Enumeration Type Documentation

#### 8.104.2.1 enum ECallbackType

the type of callback

Enumerator

***cbPostInsideLock***

***cbPostOutsideLock*** callback is fired on leaving the tree inside the lock-guarded area

### 8.104.3 Function Documentation

#### 8.104.3.1 SPINNAKER\_API void Spinnaker::GenApi::Deregister ( GenApi::CallbackHandleType *pCallbackInfo* )

Unregistering callback by handle.

#### 8.104.3.2 CNodeCallback\* Spinnaker::GenApi::make\_NodeCallback ( INode \* *pNode*, Function *function*, ECallbackType *CallbackType* )

make a new callback object for C functions

#### 8.104.3.3 CNodeCallback\* Spinnaker::GenApi::make\_NodeCallback ( INode \* *pNode*, Client & *client*, Member *member*, ECallbackType *CallbackType* )

make a new callback object for member functions

#### 8.104.3.4 intptr\_t Spinnaker::GenApi::Register ( INode \* *pNode*, Function *f*, ECallbackType *CallbackType* = *cbPostInsideLock* )

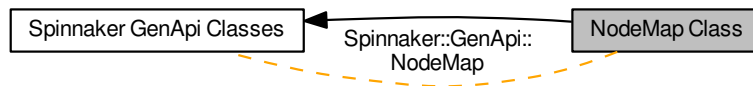
Register a C-function as a callback.

#### 8.104.3.5 intptr\_t Spinnaker::GenApi::Register ( INode \* *pNode*, Client & *c*, Member *m*, ECallbackType *CallbackType* = *cbPostInsideLock* )

Register a C++-member function a callback.

## 8.105 NodeMap Class

Collaboration diagram for NodeMap Class:



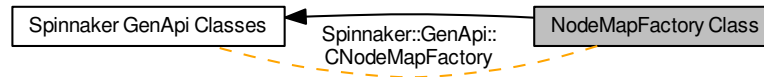
### Classes

- class [NodeMap](#)  
*Smart pointer template for NodeMaps with create function.*

### 8.105.1 Detailed Description

## 8.106 NodeMapFactory Class

Collaboration diagram for NodeMapFactory Class:



### Classes

- class [CNodeMapFactory](#)

*The node map factory is used for creating node maps from camera description files.*

### Enumerations

- enum [ECacheUsage\\_t](#) {  
[CacheUsage\\_Automatic](#),  
[CacheUsage\\_ForceWrite](#),  
[CacheUsage\\_ForceRead](#),  
[CacheUsage\\_Ignore](#) }

*Lists the cache usage strategies.*

- enum [EContentType\\_t](#) {  
[ContentType\\_Xml](#),  
[ContentType\\_ZippedXml](#) }

*Lists the processable file types.*

#### 8.106.1 Detailed Description

#### 8.106.2 Enumeration Type Documentation

##### 8.106.2.1 enum [ECacheUsage\\_t](#)

Lists the cache usage strategies.

The cache stores preprocessed camera description xml files providing faster access or smaller footprint. note The environment variable GENICAM\_CACHE\_VERSION, e.g. GENICAM\_CACHE\_V3\_0, must contain the path to cache directory for using the cache.

#### Enumerator

***CacheUsage\_Automatic*** The use of cache files is determined automatically.

***CacheUsage\_ForceWrite*** Forces the loading and preprocessing of the camera description xml file. If a cache directory is available the result of preprocessing is written to the cache.

***CacheUsage\_ForceRead*** Suppresses loading and preprocessing of the camera description xml file and forces reading a cache file from cache directory. Fails if no matching cache file is available.

***CacheUsage\_Ignore*** Forces the loading and preprocessing of the camera description xml file. No cache file is written.



## 8.106.2.2 enum EContentType\_t

Lists the processable file types.

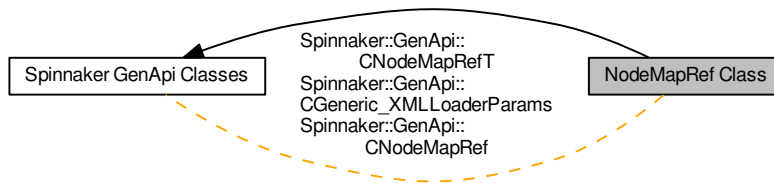
Enumerator

***ContentType\_Xml*** XML camera description file text.

***ContentType\_ZippedXml*** Zipped XML camera description file text.

## 8.107 NodeMapRef Class

Collaboration diagram for NodeMapRef Class:



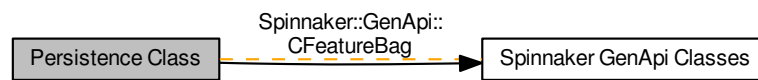
### Classes

- class [CNodeMapRefT< TCameraParams >](#)  
*Smartpointer template for NodeMaps with create function.*
- class [CGeneric\\_XMLLoaderParams](#)  
*Empty base class used by class [CNodeMapRef](#) as generic template argument.*
- class [CNodeMapRef](#)  
*Smartpointer for NodeMaps with create function.*

### 8.107.1 Detailed Description

## 8.108 Persistence Class

Collaboration diagram for Persistence Class:



### Classes

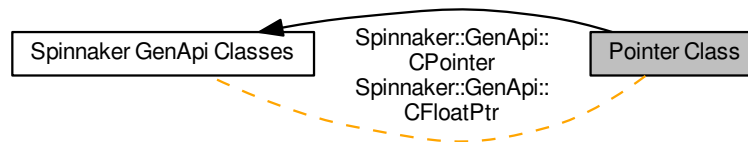
- class [CFeatureBag](#)

*Bag holding streamable features of a nodetree.*

### 8.108.1 Detailed Description

## 8.109 Pointer Class

Collaboration diagram for Pointer Class:



### Classes

- class [CPointer< T, B >](#)  
*Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.*
- class [CFloatPtr](#)  
*SmartPointer for IFloat interface pointer.*

### Typedefs

- typedef [CPointer< IBase >](#) [CBasePtr](#)  
*SmartPointer for IBase interface pointer.*
- typedef [CPointer< INode, IBase >](#) [CNodePtr](#)  
*SmartPointer for INode interface pointer.*
- typedef [CPointer< IValue >](#) [CValuePtr](#)  
*SmartPointer for IValue interface pointer.*
- typedef [CPointer< ICategory >](#) [CCategoryPtr](#)  
*SmartPointer for ICategory interface pointer.*
- typedef [CPointer< IBoolean >](#) [CBooleanPtr](#)  
*SmartPointer for IBoolean interface pointer.*
- typedef [CPointer< IInteger >](#) [CIntegerPtr](#)  
*SmartPointer for IInteger interface pointer.*
- typedef [CPointer< IString >](#) [CStringPtr](#)  
*SmartPointer for IString interface pointer.*
- typedef [CPointer< IRegister >](#) [CRegisterPtr](#)  
*SmartPointer for IRegister interface pointer.*
- typedef [CPointer< IEnumeration >](#) [CEnumerationPtr](#)  
*SmartPointer for IEnumeration interface pointer.*
- typedef [CPointer< IEnumEntry >](#) [CEnumEntryPtr](#)  
*SmartPointer for IEnumEntry interface pointer.*
- typedef [CPointer< IPort >](#) [CPortPtr](#)  
*SmartPointer for IPort interface pointer.*
- typedef [CPointer< IPortReplay >](#) [CPortReplayPtr](#)  
*SmartPointer for IPortReplay interface pointer.*
- typedef [CPointer< IPortRecorder >](#) [CPortRecorderPtr](#)

- SmartPointer for IPortRecorder interface pointer.*
- typedef CPointer< IPortWriteList, IPortWriteList > [CPortWriteListPtr](#)  
*SmartPointer for IPortWriteList interface pointer.*
- typedef CPointer< IChunkPort > [CChunkPortPtr](#)  
*SmartPointer for IChunkPort interface pointer.*
- typedef CPointer< INodeMap, INodeMap > [CNodeMapPtr](#)  
*SmartPointer for INodeMap interface pointer.*
- typedef CPointer< INodeMapDyn, INodeMap > [CNodeMapDynPtr](#)  
*SmartPointer for INodeMapDyn interface pointer.*
- typedef CPointer< IDeviceInfo, INodeMap > [CDeviceInfoPtr](#)  
*SmartPointer for IDeviceInfo interface pointer.*
- typedef CPointer< ISelector > [CSelectorPtr](#)  
*SmartPointer for ISelector interface pointer.*
- typedef CPointer< ICommand > [CCommandPtr](#)  
*SmartPointer for ICommand interface pointer.*
- typedef CPointer< IPortConstruct > [CPortConstructPtr](#)  
*SmartPointer for IPortConstruct interface pointer.*

## Functions

- template<class T , class B >  
bool [IsReadable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)  
*Checks if a node is readable.*
- template<class T , class B >  
bool [IsWritable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)  
*Checks if a node is Writable.*
- template<class T , class B >  
bool [IsImplemented](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)  
*Checks if a node is Implemented.*
- template<class T , class B >  
bool [IsAvailable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)  
*Checks if a node is Available.*
- GenICam::gcstring [GetInterfaceName](#) (IBase \*pBase)  
*Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.*

### 8.109.1 Detailed Description

### 8.109.2 Typedef Documentation

#### 8.109.2.1 typedef CPointer<IBase> CBasePtr

SmartPointer for IBase interface pointer.

#### 8.109.2.2 typedef CPointer<IBoolean> CBooleanPtr

SmartPointer for IBoolean interface pointer.

**8.109.2.3    typedef CPointer<ICategory> CCategoryPtr**

SmartPointer for ICategory interface pointer.

**8.109.2.4    typedef CPointer<IChunkPort> CChunkPortPtr**

SmartPointer for IChunkPort interface pointer.

**8.109.2.5    typedef CPointer<ICommand> CCommandPtr**

SmartPointer for ICommand interface pointer.

**8.109.2.6    typedef CPointer<IDeviceInfo, INodeMap> CDeviceInfoPtr**

SmartPointer for IDeviceInfo interface pointer.

**8.109.2.7    typedef CPointer<IEnumEntry> CEnumEntryPtr**

SmartPointer for IEnumEntry interface pointer.

**8.109.2.8    typedef CPointer<IEnumeration> CEnumerationPtr**

SmartPointer for IEnumeration interface pointer.

**8.109.2.9    typedef CPointer<IInteger> CIntegerPtr**

SmartPointer for IInteger interface pointer.

**8.109.2.10    typedef CPointer<INodeMapDyn, INodeMap> CNodeMapDynPtr**

SmartPointer for INodeMapDyn interface pointer.

**8.109.2.11    typedef CPointer<INodeMap, INodeMap> CNodeMapPtr**

SmartPointer for INodeMap interface pointer.

**8.109.2.12    typedef CPointer<INode, IBase> CNodePtr**

SmartPointer for INode interface pointer.

8.109.2.13 `typedef CPointer<IPortConstruct> CPortConstructPtr`

SmartPointer for IPortConstruct interface pointer.

8.109.2.14 `typedef CPointer<IPort> CPortPtr`

SmartPointer for IPort interface pointer.

8.109.2.15 `typedef CPointer<IPortRecorder> CPortRecorderPtr`

SmartPointer for IPortRecorder interface pointer.

8.109.2.16 `typedef CPointer<IPortReplay> CPortReplayPtr`

SmartPointer for IPortReplay interface pointer.

8.109.2.17 `typedef CPointer<IPortWriteList, IPortWriteList> CPortWriteListPtr`

SmartPointer for IPortWriteList interface pointer.

8.109.2.18 `typedef CPointer<IRegister> CRegisterPtr`

SmartPointer for IRegister interface pointer.

8.109.2.19 `typedef CPointer<ISelector> CSelectorPtr`

SmartPointer for ISelector interface pointer.

8.109.2.20 `typedef CPointer<IString> CStringPtr`

SmartPointer for IString interface pointer.

8.109.2.21 `typedef CPointer<IValue> CValuePtr`

SmartPointer for IValue interface pointer.

### 8.109.3 Function Documentation

8.109.3.1 `GenICam::gcstring Spinnaker::GenApi::GetInterfaceName ( IBase * pBase ) [inline]`

Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.

**8.109.3.2** `bool Spinnaker::GenApi::IsAvailable ( const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]`

Checks if a node is Available.

**8.109.3.3** `bool Spinnaker::GenApi::IsImplemented ( const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]`

Checks if a node is Implemented.

**8.109.3.4** `bool Spinnaker::GenApi::IsReadable ( const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]`

Checks if a node is readable.

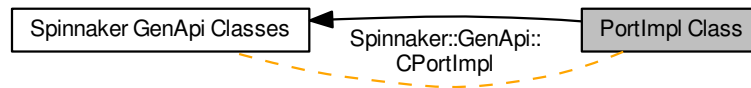
**8.109.3.5** `bool Spinnaker::GenApi::IsWritable ( const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]`

Checks if a node is Writable.



## 8.110 PortImpl Class

Collaboration diagram for PortImpl Class:



### Classes

- class [CPortImpl](#)  
*Standard implementation for a port.*

### 8.110.1 Detailed Description

## 8.111 PortNode Class

Collaboration diagram for PortNode Class:



### Classes

- class [PortNode](#)  
*[Interface](#) for value properties.*

### Typedefs

- typedef PortNode [CPortRef](#)

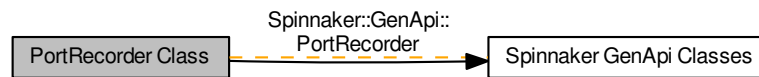
#### 8.111.1 Detailed Description

#### 8.111.2 Typedef Documentation

##### 8.111.2.1 typedef PortNode CPortRef

## 8.112 PortRecorder Class

Collaboration diagram for PortRecorder Class:



### Classes

- class [PortRecorder](#)  
*Interface for recording write commands on a port.*

### Typedefs

- typedef PortRecorder [CPortRecorderRef](#)  
*Reference to an IPortRecorder pointer.*

#### 8.112.1 Detailed Description

#### 8.112.2 Typedef Documentation

##### 8.112.2.1 typedef PortRecorder CPortRecorderRef

Reference to an IPortRecorder pointer.

## 8.113 PortReplay Class

Collaboration diagram for PortReplay Class:



### Classes

- class [PortReplay](#)  
*Interface for replaying write commands on a port.*

### 8.113.1 Detailed Description

## 8.114 PortWriteList Class

Collaboration diagram for PortWriteList Class:



### Classes

- class [CPortWriteList](#)  
*Container holding a list of port write commands.*

### 8.114.1 Detailed Description

## 8.115 Reference Interfaces

Collaboration diagram for Reference Interfaces:



### Functions

- virtual void [SetNumEnums](#) (int NumEnums)=0  
*sets the number of enum values*

#### 8.115.1 Detailed Description

#### 8.115.2 Function Documentation

8.115.2.1 virtual void Spinnaker::GenApi::SetNumEnums ( int *NumEnums* ) [pure virtual]

sets the number of enum values

## 8.116 RegisterNode Class

Collaboration diagram for RegisterNode Class:



### Classes

- class [RegisterNode](#)  
*[Interface](#) for string properties.*

### Typedefs

- typedef RegisterNode [CRegisterRef](#)

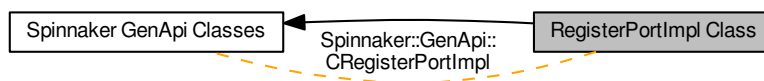
#### 8.116.1 Detailed Description

#### 8.116.2 Typedef Documentation

##### 8.116.2.1 typedef RegisterNode CRegisterRef

## 8.117 RegisterPortImpl Class

Collaboration diagram for RegisterPortImpl Class:



### Classes

- class [CRegisterPortImpl](#)

*Standard implementation for a port using a register based transport layer.*

### 8.117.1 Detailed Description



## 8.118 SelectorSet Class

Collaboration diagram for SelectorSet Class:



### Classes

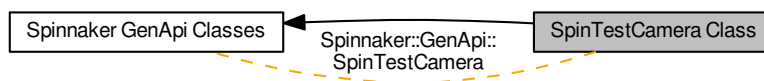
- class [CSelectorSet](#)

*The set of selectors selecting a given node.*

### 8.118.1 Detailed Description

## 8.119 SpinTestCamera Class

Collaboration diagram for SpinTestCamera Class:



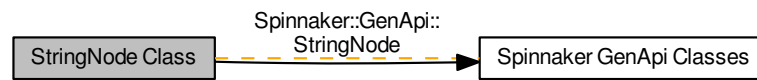
### Classes

- class [SpinTestCamera](#)

### 8.119.1 Detailed Description

## 8.120 StringNode Class

Collaboration diagram for StringNode Class:



### Classes

- class [StringNode](#)  
*[Interface](#) for string properties.*

### Typedefs

- typedef StringNode [CStringRef](#)

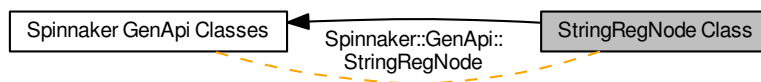
#### 8.120.1 Detailed Description

#### 8.120.2 Typedef Documentation

##### 8.120.2.1 typedef StringNode CStringRef

## 8.121 StringRegNode Class

Collaboration diagram for StringRegNode Class:



### Classes

- class [StringRegNode](#)  
*Interface for string properties.*

#### 8.121.1 Detailed Description

## 8.122 StructPort Class

Collaboration diagram for StructPort Class:



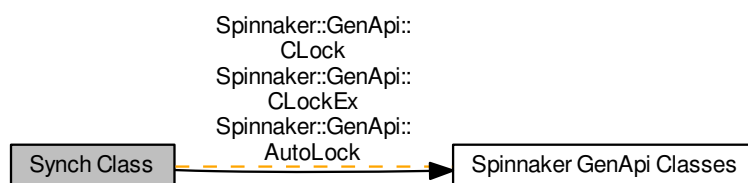
### Classes

- class `CTestPortStruct< CDataStruct >`  
*Implements a register spaces based on a C++ struct.*

### 8.122.1 Detailed Description

## 8.123 Synch Class

Collaboration diagram for Synch Class:



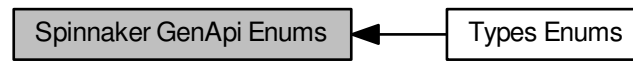
### Classes

- class [CLock](#)  
*A lock class.*
- class [CLockEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)

### 8.123.1 Detailed Description

## 8.124 Spinnaker GenApi Enums

Collaboration diagram for Spinnaker GenApi Enums:



### Modules

- [Types Enums](#)

### 8.124.1 Detailed Description

## 8.125 Types Enums

Collaboration diagram for Types Enums:



### Macros

- `#define \_UndefinedRepresentation _UndefinedRepresentation`

### Typedefs

- `typedef GenICam::gcstring_vector StringList\_t`  
*A list of strings.*

### Enumerations

- `enum ESign {`  
`Signed,`  
`Unsigned,`  
`\_UndefinedSign }`  
*signed or unsigned integers*
- `enum EAccessMode {`  
`NI,`  
`NA,`  
`WO,`  
`RO,`  
`RW,`  
`\_UndefinedAccesMode,`  
`\_CycleDetectAccesMode }`  
*access mode of a node*
- `enum EVisibility {`  
`Beginner = 0,`  
`Expert = 1,`  
`Guru = 2,`  
`Invisible = 3,`  
`\_UndefinedVisibility = 99 }`  
*recommended visibility of a node*
- `enum ECachingMode {`  
`NoCache,`  
`WriteThrough,`  
`WriteAround,`  
`\_UndefinedCachingMode }`  
*caching mode of a register*



- enum [ERepresentation](#) {  
[Linear](#),  
[Logarithmic](#),  
[Boolean](#),  
[PureNumber](#),  
[HexNumber](#),  
[IPV4Address](#),  
[MACAddress](#),  
[\\_UndefinedRepresentation](#) }  
*recommended representation of a node value*
- enum [EEndianness](#) {  
[BigEndian](#),  
[LittleEndian](#),  
[\\_UndefinedEndian](#) }  
*Endianness of a value in a register.*
- enum [ENameSpace](#) {  
[Custom](#),  
[Standard](#),  
[\\_UndefinedNameSpace](#) }  
*Defines if a node name is standard or custom.*
- enum [EStandardNameSpace](#) {  
[None](#),  
[GEV](#),  
[IIDC](#),  
[CL](#),  
[USB](#),  
[\\_UndefinedStandardNameSpace](#) }  
*Defines from which standard namespace a node name comes from.*
- enum [EYesNo](#) {  
[Yes](#) = 1,  
[No](#) = 0,  
[\\_UndefinedYesNo](#) = 2 }  
*Defines the choices of a Yes/No alternative.*
- enum [ESlope](#) {  
[Increasing](#),  
[Decreasing](#),  
[Varying](#),  
[Automatic](#),  
[\\_UndefinedESlope](#) }  
*typedef for formula type*
- enum [EXMLValidation](#) {  
[xvLoad](#) = 0x00000001L,  
[xvCycles](#) = 0x00000002L,  
[xvSFNC](#) = 0x00000004L,  
[xvDefault](#) = 0x00000000L,  
[xvAll](#) = 0xffffffffL,  
[\\_UndefinedEXMLValidation](#) = 0x80000000L }  
*typedef describing the different validity checks which can be performed on an XML file*
- enum [EDisplayNotation](#) {  
[fnAutomatic](#),  
[fnFixed](#),  
[fnScientific](#),  
[\\_UndefinedEDisplayNotation](#) }  
*typedef for float notation*

- enum `EInterfaceType` {  
`intflValue`,  
`intflBase`,  
`intflInteger`,  
`intflBoolean`,  
`intflCommand`,  
`intflFloat`,  
`intflString`,  
`intflRegister`,  
`intflCategory`,  
`intflEnumeration`,  
`intflEnumEntry`,  
`intflPort` }  
*typedef for interface type*
- enum `ELinkType` {  
`ctParentNodes`,  
`ctReadingChildren`,  
`ctWritingChildren`,  
`ctInvalidatingChildren`,  
`ctDependingNodes`,  
`ctTerminalNodes` }  
*typedef for link type*
- enum `ELncMode` {  
`noIncrement`,  
`fixedIncrement`,  
`listIncrement` }  
*typedef for increment mode*
- enum `EInputDirection` {  
`idFrom`,  
`idTo`,  
`idNone` }  
*typedef for link type*
- enum `EGenApiSchemaVersion` {  
`v1_0` = 1,  
`v1_1` = 2,  
`_Undefined` = -1 }  
*GenApi schema version.*

### 8.125.1 Detailed Description

### 8.125.2 Macro Definition Documentation

#### 8.125.2.1 `#define _UndefinedRepresentation _UndefinedRepresentation`

### 8.125.3 Typedef Documentation

#### 8.125.3.1 `typedef GenlCam::gcstring_vector StringList_t`

A list of strings.

### 8.125.4 Enumeration Type Documentation

#### 8.125.4.1 enum EAccessMode

access mode of a node

Enumerator

***NI***

***NA*** Not implemented.

***WO*** Not available.

***RO*** Write Only.

***RW*** Read Only.

***\_UndefinedAccesMode*** Read and Write.

***\_CycleDetectAccesMode*** Object is not yet initialized. used internally for AccessMode cycle detection

#### 8.125.4.2 enum ECachingMode

caching mode of a register

Enumerator

***NoCache***

***WriteThrough*** Do not use cache.

***WriteAround*** Write to cache and register.

***\_UndefinedCachingMode*** Write to register, write to cache on read. Not yet initialized

#### 8.125.4.3 enum EDisplayNotation

typedef for float notation

Enumerator

***fnAutomatic***

***fnFixed*** the notation if either scientific or fixed depending on what is shorter

***fnScientific*** the notation is fixed, e.g. 123.4

***\_UndefinedEDisplayNotation*** the notation is scientific, e.g. 1.234e2 Object is not yet initialized

#### 8.125.4.4 enum EEndianess

Endianess of a value in a register.

Enumerator

***BigEndian***

***LittleEndian*** Register is big endian.

***\_UndefinedEndian*** Register is little endian. Object is not yet initialized

## 8.125.4.5 enum EGenApiSchemaVersion

GenApi schema version.

Enumerator

***v1\_0***  
***v1\_1***  
***\_Undefined***

## 8.125.4.6 enum EIncMode

typedef for increment mode

Enumerator

***noIncrement***  
***fixedIncrement*** The feature has no increment.  
***listIncrement*** The feature has a fix increment.

## 8.125.4.7 enum EInputDirection

typedef for link type

Enumerator

***idFrom***  
***idTo*** Indicates a swiss knife that it is used as worker for a converter computing FROM.  
***idNone*** Indicates a swiss knife that it is used as worker for a converter computing TO. SwissKnife is not used within a converter

## 8.125.4.8 enum EInterfaceType

typedef for interface type

Enumerator

***intflValue***  
***intflBase*** IValue interface.  
***intflInteger*** IBase interface.  
***intflBoolean*** IInteger interface.  
***intflCommand*** IBoolean interface.  
***intflFloat*** ICommand interface.  
***intflString*** IFloat interface.  
***intflRegister*** IString interface.  
***intflCategory*** IRegister interface.  
***intflEnumeration*** ICategory interface.  
***intflEnumEntry*** IEnumeration interface.  
***intflPort*** IEnumEntry interface. IPort interface

## 8.125.4.9 enum ELinkType

typedef for link type

Enumerator

**ctParentNodes**

**ctReadingChildren** All nodes for which this node is at least an invalidating child.

**ctWritingChildren** All nodes which can be read from.

**ctInvalidatingChildren** All nodes which can write a value further down the node stack.

**ctDependingNodes** All directly connected nodes which invalidate this node.

**ctTerminalNodes** All directly or indirectly connected nodes which are invalidated by this nodes (i.e. which are dependent on this node) All indirectly connected terminal nodes

## 8.125.4.10 enum ENameSpace

Defines if a node name is standard or custom.

Enumerator

**Custom**

**Standard** name resides in custom namespace

**\_UndefinedNameSpace** name resides in one of the standard namespaces Object is not yet initialized

## 8.125.4.11 enum ERepresentation

recommended representation of a node value

Enumerator

**Linear**

**Logarithmic** Slider with linear behavior.

**Boolean** Slider with logarithmic behavior.

**PureNumber** Check box.

**HexNumber** Decimal number in an edit control.

**IPV4Address** Hex number in an edit control.

**MACAddress** IP-Address.

**\_UndefinedRepresentation** MAC-Address.

## 8.125.4.12 enum ESign

signed or unsigned integers

Enumerator

**Signed**

**Unsigned** Integer is signed.

**\_UndefinedSign** Integer is unsigned. Object is not yet initialized

## 8.125.4.13 enum ESlope

typedef for formula type

Enumerator

**Increasing**

**Decreasing** strictly monotonous increasing

**Varying** strictly monotonous decreasing

**Automatic** slope changes, e.g. at run-time

**\_UndefinedESlope** slope is determined automatically by probing the function Object is not yet initialized

## 8.125.4.14 enum EStandardNameSpace

Defines from which standard namespace a node name comes from.

Enumerator

**None**

**GEV** name resides in custom namespace

**IIDC** name resides in GigE Vision namespace

**CL** name resides in 1394 IIDC namespace

**USB** name resides in camera link namespace

**\_UndefinedStandardNameSpace** name resides in USB namespace Object is not yet initialized

## 8.125.4.15 enum EVisibility

recommended visibility of a node

Enumerator

**Beginner**

**Expert** Always visible.

**Guru** Visible for experts or Gurus.

**Invisible** Visible for Gurus.

**\_UndefinedVisibility** Not Visible.

## 8.125.4.16 enum EXMLValidation

typedef describing the different validity checks which can be performed on an XML file

The enum values for a bit field of length uint32\_t

Enumerator

**xvLoad**

**xvCycles** Creates a dummy node map.

**xvSFNC** checks for write and dependency cycles (implies xvLoad)

**xvDefault** checks for conformance with the standard feature naming convention (SFNC)

**xvAll** checks performed if nothing else is said

**\_UndefinedEXMLValidation** all possible checks

## 8.125.4.17 enum EYesNo

Defines the choices of a Yes/No alternative.

Enumerator

**Yes**

**No** yes

**\_UndefinedYesNo** no

## 8.126 ValueNode Class

Collaboration diagram for ValueNode Class:



### Classes

- class [ValueNode](#)  
*[Interface](#) for value properties.*

### Typedefs

- typedef ValueNode [CValueRef](#)

#### 8.126.1 Detailed Description

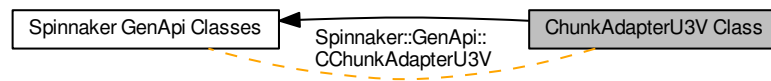
#### 8.126.2 Typedef Documentation

##### 8.126.2.1 typedef ValueNode CValueRef



## 8.127 ChunkAdapterU3V Class

Collaboration diagram for ChunkAdapterU3V Class:



### Classes

- class [CChunkAdapterU3V](#)  
*Connects a chunked U3V buffer to a node map.*

#### 8.127.1 Detailed Description



## Chapter 9

# Namespace Documentation

### 9.1 Spinnaker Namespace Reference

#### Namespaces

- [GenApi](#)
- [GenICam](#)
- [Video](#)

#### Classes

- struct [ActionCommandResult](#)  
*Action Command Result.*
- class [ArrivalEvent](#)  
*An event handler for capturing the device arrival event.*
- class [BasePtr](#)  
*The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.*
- struct [BMPOption](#)  
*Options for saving Bitmap image.*
- class [Camera](#)  
*The camera object class.*
- class [CameraBase](#)  
*The base class for the camera object.*
- class [CameraList](#)  
*Used to hold a list of camera objects.*
- class [CameraPtr](#)  
*A reference tracked pointer to a camera object.*
- class [ChunkData](#)  
*The chunk data which contains additional information about an image.*
- class [DeviceEvent](#)  
*A handler to device events.*
- class [Event](#)  
*The base class for all event types.*
- class [Exception](#)  
*The [Exception](#) object represents an error that is returned from the library.*

- class [IArrivalEvent](#)
- class [ICameraBase](#)
  - The interface file for base class for the camera object.*
- class [ICameraList](#)
  - Used to hold a list of camera objects.*
- class [IChunkData](#)
  - The [Interface](#) file for [ChunkData](#).*
- class [IDeviceEvent](#)
- class [IImage](#)
  - The interface file for [Image](#).*
- class [IImageEvent](#)
- class [IImageStatistics](#)
  - The interface file for image statistics.*
- class [IInterface](#)
  - The interface file for [Interface](#).*
- class [IInterfaceEvent](#)
- class [IInterfaceList](#)
  - The interface file for [InterfaceList](#) class.*
- class [ILoggingEvent](#)
- class [Image](#)
  - The image object class.*
- class [ImageEvent](#)
  - A handler for capturing image arrival events.*
- class [ImagePtr](#)
  - A reference tracked pointer to an image object.*
- class [ImageStatistics](#)
  - Represents image statistics for an image.*
- class [Interface](#)
  - An interface object which holds a list of cameras.*
- class [InterfaceEvent](#)
  - A handler to device arrival and removal events on all interfaces.*
- class [InterfaceList](#)
  - A list of the available interfaces on the system.*
- class [InterfacePtr](#)
  - A reference tracked pointer to the interface object.*
- class [IRemovalEvent](#)
- class [ISystem](#)
  - The interface file for [System](#).*
- struct [JPEGOption](#)
  - Options for saving JPEG image.*
- struct [JPG2Option](#)
  - Options for saving JPEG2000 image.*
- struct [LibraryVersion](#)
  - Provides easier access to the current version of [Spinnaker](#).*
- class [LoggingEvent](#)
  - An event handler for capturing the device logging event.*
- class [LoggingEventData](#)
  - The [LoggingEventData](#) object.*
- class [LoggingEventDataPtr](#)
  - A reference tracked pointer to the [LoggingEvent](#) object.*
- struct [PGMOption](#)

- Options for saving PGM images.
- struct [PNGOption](#)
  - Options for saving PNG images.
- struct [PPMOption](#)
  - Options for saving PPM images.
- class [RemovalEvent](#)
  - An event handler for capturing the device removal event.
- class [System](#)
  - The system object is used to retrieve the list of interfaces and cameras available.
- class [SystemPtr](#)
  - A reference tracked pointer to a system object.
- struct [TIFFOption](#)
  - Options for saving TIFF images.
- class [TransportLayerDevice](#)
  - Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.
- class [TransportLayerInterface](#)
  - Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.
- class [TransportLayerStream](#)
  - Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

## Enumerations

- enum [LUTSelectorEnums](#) {  
[LUTSelector\\_LUT1](#),  
[NUM\\_LUTSELECTOR](#) }
  - The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.
- enum [ExposureModeEnums](#) {  
[ExposureMode\\_Timed](#),  
[ExposureMode\\_TriggerWidth](#),  
[NUM\\_EXPOSUREMODE](#) }
- enum [AcquisitionModeEnums](#) {  
[AcquisitionMode\\_Continuous](#),  
[AcquisitionMode\\_SingleFrame](#),  
[AcquisitionMode\\_MultiFrame](#),  
[NUM\\_ACQUISITIONMODE](#) }
- enum [TriggerSourceEnums](#) {  
[TriggerSource\\_Software](#),  
[TriggerSource\\_Line0](#),  
[TriggerSource\\_Line1](#),  
[TriggerSource\\_Line2](#),  
[TriggerSource\\_Line3](#),  
[TriggerSource\\_UserOutput0](#),  
[TriggerSource\\_UserOutput1](#),  
[TriggerSource\\_UserOutput2](#),  
[TriggerSource\\_UserOutput3](#),  
[TriggerSource\\_Counter0Start](#),  
[TriggerSource\\_Counter1Start](#),  
[TriggerSource\\_Counter0End](#),  
[TriggerSource\\_Counter1End](#),  
[TriggerSource\\_LogicBlock0](#),  
[TriggerSource\\_LogicBlock1](#),  
[TriggerSource\\_Action0](#),  
[NUM\\_TRIGGERSOURCE](#) }

- enum [TriggerActivationEnums](#) {  
    [TriggerActivation\\_LevelLow](#),  
    [TriggerActivation\\_LevelHigh](#),  
    [TriggerActivation\\_FallingEdge](#),  
    [TriggerActivation\\_RisingEdge](#),  
    [TriggerActivation\\_AnyEdge](#),  
    [NUM\\_TRIGGERACTIVATION](#) }
- enum [SensorShutterModeEnums](#) {  
    [SensorShutterMode\\_Global](#),  
    [SensorShutterMode\\_Rolling](#),  
    [SensorShutterMode\\_GlobalReset](#),  
    [NUM\\_SENSORSHUTTERMODE](#) }
- enum [TriggerModeEnums](#) {  
    [TriggerMode\\_Off](#),  
    [TriggerMode\\_On](#),  
    [NUM\\_TRIGGERMODE](#) }
- enum [TriggerOverlapEnums](#) {  
    [TriggerOverlap\\_Off](#),  
    [TriggerOverlap\\_ReadOut](#),  
    [TriggerOverlap\\_PreviousFrame](#),  
    [NUM\\_TRIGGEROVERLAP](#) }
- enum [TriggerSelectorEnums](#) {  
    [TriggerSelector\\_AcquisitionStart](#),  
    [TriggerSelector\\_FrameStart](#),  
    [TriggerSelector\\_FrameBurstStart](#),  
    [NUM\\_TRIGGERSELECTOR](#) }
- enum [ExposureAutoEnums](#) {  
    [ExposureAuto\\_Off](#),  
    [ExposureAuto\\_Once](#),  
    [ExposureAuto\\_Continuous](#),  
    [NUM\\_EXPOSUREAUTO](#) }
- enum [EventSelectorEnums](#) {  
    [EventSelector\\_Error](#),  
    [EventSelector\\_ExposureEnd](#),  
    [EventSelector\\_SerialPortReceive](#),  
    [NUM\\_EVENTSELECTOR](#) }
- enum [EventNotificationEnums](#) {  
    [EventNotification\\_On](#),  
    [EventNotification\\_Off](#),  
    [NUM\\_EVENTNOTIFICATION](#) }
- enum [LogicBlockSelectorEnums](#) {  
    [LogicBlockSelector\\_LogicBlock0](#),  
    [LogicBlockSelector\\_LogicBlock1](#),  
    [NUM\\_LOGICBLOCKSELECTOR](#) }
- enum [LogicBlockLUTInputActivationEnums](#) {  
    [LogicBlockLUTInputActivation\\_LevelLow](#),  
    [LogicBlockLUTInputActivation\\_LevelHigh](#),  
    [LogicBlockLUTInputActivation\\_FallingEdge](#),  
    [LogicBlockLUTInputActivation\\_RisingEdge](#),  
    [LogicBlockLUTInputActivation\\_AnyEdge](#),  
    [NUM\\_LOGICBLOCKLUTINPUTACTIVATION](#) }
- enum [LogicBlockLUTInputSelectorEnums](#) {  
    [LogicBlockLUTInputSelector\\_Input0](#),  
    [LogicBlockLUTInputSelector\\_Input1](#),  
    [LogicBlockLUTInputSelector\\_Input2](#),  
    [LogicBlockLUTInputSelector\\_Input3](#),  
    [NUM\\_LOGICBLOCKLUTINPUTSELECTOR](#) }

- enum [LogicBlockLUTInputSourceEnums](#) {  
[LogicBlockLUTInputSource\\_Zero](#),  
[LogicBlockLUTInputSource\\_Line0](#),  
[LogicBlockLUTInputSource\\_Line1](#),  
[LogicBlockLUTInputSource\\_Line2](#),  
[LogicBlockLUTInputSource\\_Line3](#),  
[LogicBlockLUTInputSource\\_UserOutput0](#),  
[LogicBlockLUTInputSource\\_UserOutput1](#),  
[LogicBlockLUTInputSource\\_UserOutput2](#),  
[LogicBlockLUTInputSource\\_UserOutput3](#),  
[LogicBlockLUTInputSource\\_Counter0Start](#),  
[LogicBlockLUTInputSource\\_Counter1Start](#),  
[LogicBlockLUTInputSource\\_Counter0End](#),  
[LogicBlockLUTInputSource\\_Counter1End](#),  
[LogicBlockLUTInputSource\\_LogicBlock0](#),  
[LogicBlockLUTInputSource\\_LogicBlock1](#),  
[LogicBlockLUTInputSource\\_ExposureStart](#),  
[LogicBlockLUTInputSource\\_ExposureEnd](#),  
[LogicBlockLUTInputSource\\_FrameTriggerWait](#),  
[LogicBlockLUTInputSource\\_AcquisitionActive](#),  
[NUM\\_LOGICBLOCKLUTINPUTSOURCE](#) }
- enum [LogicBlockLUTSelectorEnums](#) {  
[LogicBlockLUTSelector\\_Value](#),  
[LogicBlockLUTSelector\\_Enable](#),  
[NUM\\_LOGICBLOCKLUTSELECTOR](#) }
- enum [ColorTransformationSelectorEnums](#) {  
[ColorTransformationSelector\\_RGBtoRGB](#),  
[ColorTransformationSelector\\_RGBtoYUV](#),  
[NUM\\_COLORTRANSFORMATIONSELECTOR](#) }
- enum [RgbTransformLightSourceEnums](#) {  
[RgbTransformLightSource\\_General](#),  
[RgbTransformLightSource\\_Tungsten2800K](#),  
[RgbTransformLightSource\\_WarmFluorescent3000K](#),  
[RgbTransformLightSource\\_CoolFluorescent4000K](#),  
[RgbTransformLightSource\\_Daylight5000K](#),  
[RgbTransformLightSource\\_Cloudy6500K](#),  
[RgbTransformLightSource\\_Shade8000K](#),  
[RgbTransformLightSource\\_Custom](#),  
[NUM\\_RGBTRANSFORMLIGHTSOURCE](#) }
- enum [ColorTransformationValueSelectorEnums](#) {  
[ColorTransformationValueSelector\\_Gain00](#),  
[ColorTransformationValueSelector\\_Gain01](#),  
[ColorTransformationValueSelector\\_Gain02](#),  
[ColorTransformationValueSelector\\_Gain10](#),  
[ColorTransformationValueSelector\\_Gain11](#),  
[ColorTransformationValueSelector\\_Gain12](#),  
[ColorTransformationValueSelector\\_Gain20](#),  
[ColorTransformationValueSelector\\_Gain21](#),  
[ColorTransformationValueSelector\\_Gain22](#),  
[ColorTransformationValueSelector\\_Offset0](#),  
[ColorTransformationValueSelector\\_Offset1](#),  
[ColorTransformationValueSelector\\_Offset2](#),  
[NUM\\_COLORTRANSFORMATIONVALUESELECTOR](#) }
- enum [DeviceRegistersEndiannessEnums](#) {  
[DeviceRegistersEndianness\\_Little](#),  
[DeviceRegistersEndianness\\_Big](#),  
[NUM\\_DEVICEREGISTERSENDIANNES](#) }
- enum [DeviceScanTypeEnums](#) {

- DeviceScanType\_Areascan,  
NUM\_DEVICESCANTYPE }
- enum DeviceCharacterSetEnums {  
DeviceCharacterSet\_UTF8,  
DeviceCharacterSet\_ASCII,  
NUM\_DEVICECHARACTERSET }
- enum DeviceTLTypeEnums {  
DeviceTLType\_GigEVision,  
DeviceTLType\_CameraLink,  
DeviceTLType\_CameraLinkHS,  
DeviceTLType\_CoaXPress,  
DeviceTLType\_USB3Vision,  
DeviceTLType\_Custom,  
NUM\_DEVICETLTYPE }
- enum DevicePowerSupplySelectorEnums {  
DevicePowerSupplySelector\_External,  
NUM\_DEVICEPOWERSUPPLYSELECTOR }
- enum DeviceTemperatureSelectorEnums {  
DeviceTemperatureSelector\_Sensor,  
NUM\_DEVICETEMPERATURESELECTOR }
- enum DeviceIndicatorModeEnums {  
DeviceIndicatorMode\_Inactive,  
DeviceIndicatorMode\_Active,  
DeviceIndicatorMode\_ErrorStatus,  
NUM\_DEVICEINDICATORMODE }
- enum AutoExposureControlPriorityEnums {  
AutoExposureControlPriority\_Gain,  
AutoExposureControlPriority\_ExposureTime,  
NUM\_AUTOEXPOSURECONTROLPRIORITY }
- enum AutoExposureMeteringModeEnums {  
AutoExposureMeteringMode\_Average,  
AutoExposureMeteringMode\_Spot,  
AutoExposureMeteringMode\_Partial,  
AutoExposureMeteringMode\_CenterWeighted,  
AutoExposureMeteringMode\_HistogramPeak,  
NUM\_AUTOEXPOSUREMETERINGMODE }
- enum BalanceWhiteAutoProfileEnums {  
BalanceWhiteAutoProfile\_Indoor,  
BalanceWhiteAutoProfile\_Outdoor,  
NUM\_BALANCEWHITEAUTOPROFILE }
- enum AutoAlgorithmSelectorEnums {  
AutoAlgorithmSelector\_Awb,  
AutoAlgorithmSelector\_Ae,  
NUM\_AUTOALGORITHMSELECTOR }
- enum AutoExposureTargetGreyValueAutoEnums {  
AutoExposureTargetGreyValueAuto\_Off,  
AutoExposureTargetGreyValueAuto\_Continuous,  
NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO }
- enum AutoExposureLightingModeEnums {  
AutoExposureLightingMode\_AutoDetect,  
AutoExposureLightingMode\_Backlight,  
AutoExposureLightingMode\_Frontlight,  
AutoExposureLightingMode\_Normal,  
NUM\_AUTOEXPOSURELIGHTINGMODE }
- enum GevIEEE1588StatusEnums {



- ```

GevIEEE1588Status_Initializing,
GevIEEE1588Status_Faulty,
GevIEEE1588Status_Disabled,
GevIEEE1588Status_Listening,
GevIEEE1588Status_PreMaster,
GevIEEE1588Status_Master,
GevIEEE1588Status_Passive,
GevIEEE1588Status_Uncalibrated,
GevIEEE1588Status_Slave,
NUM_GEVIEEE1588STATUS }

```
- enum GevIEEE1588ModeEnums {

```

GevIEEE1588Mode_Auto,
GevIEEE1588Mode_SlaveOnly,
NUM_GEVIEEE1588MODE }

```
  - enum GevIEEE1588ClockAccuracyEnums {

```

GevIEEE1588ClockAccuracy_Unknown,
NUM_GEVIEEE1588CLOCKACCURACY }

```
  - enum GevCCPEnums {

```

GevCCP_OpenAccess,
GevCCP_ExclusiveAccess,
GevCCP_ControlAccess,
NUM_GEVCCP }

```
  - enum GevSupportedOptionSelectorEnums {

```

GevSupportedOptionSelector_UserDefinedName,
GevSupportedOptionSelector_SerialNumber,
GevSupportedOptionSelector_HeartbeatDisable,
GevSupportedOptionSelector_LinkSpeed,
GevSupportedOptionSelector_CCPApplicationSocket,
GevSupportedOptionSelector_ManifestTable,
GevSupportedOptionSelector_TestData,
GevSupportedOptionSelector_DiscoveryAckDelay,
GevSupportedOptionSelector_DiscoveryAckDelayWritable,
GevSupportedOptionSelector_ExtendedStatusCodes,
GevSupportedOptionSelector_Action,
GevSupportedOptionSelector_PendingAck,
GevSupportedOptionSelector_EventData,
GevSupportedOptionSelector_Event,
GevSupportedOptionSelector_PacketResend,
GevSupportedOptionSelector_WriteMem,
GevSupportedOptionSelector_CommandsConcatenation,
GevSupportedOptionSelector_IPConfigurationLLA,
GevSupportedOptionSelector_IPConfigurationDHCP,
GevSupportedOptionSelector_IPConfigurationPersistentIP,
GevSupportedOptionSelector_StreamChannelSourceSocket,
GevSupportedOptionSelector_MessageChannelSourceSocket,
NUM_GEVSUPPORTEDOPTIONSELECTOR }

```
  - enum BlackLevelSelectorEnums {

```

BlackLevelSelector_All,
BlackLevelSelector_Analog,
BlackLevelSelector_Digital,
NUM_BLACKLEVELSELECTOR }

```
  - enum BalanceWhiteAutoEnums {

```

BalanceWhiteAuto_Off,
BalanceWhiteAuto_Once,
BalanceWhiteAuto_Continuous,
NUM_BALANCEWHITEAUTO }

```
  - enum GainAutoEnums {

```
GainAuto_Off,  
GainAuto_Once,  
GainAuto_Continuous,  
NUM_GAINAUTO }  
  
• enum BalanceRatioSelectorEnums {  
    BalanceRatioSelector_Red,  
    BalanceRatioSelector_Blue,  
    NUM_BALANCERATIOSELECTOR }  
  
• enum GainSelectorEnums {  
    GainSelector_All,  
    NUM_GAINSELECTOR }  
  
• enum DefectCorrectionModeEnums {  
    DefectCorrectionMode_Average,  
    DefectCorrectionMode_Highlight,  
    DefectCorrectionMode_Zero,  
    NUM_DEFECTCORRECTIONMODE }  
  
• enum UserSetSelectorEnums {  
    UserSetSelector_Default,  
    UserSetSelector_UserSet0,  
    UserSetSelector_UserSet1,  
    NUM_USERSETSELECTOR }  
  
• enum UserSetDefaultEnums {  
    UserSetDefault_Default,  
    UserSetDefault_UserSet0,  
    UserSetDefault_UserSet1,  
    NUM_USERSETDEFAULT }  
  
• enum SerialPortBaudRateEnums {  
    SerialPortBaudRate_Baud300,  
    SerialPortBaudRate_Baud600,  
    SerialPortBaudRate_Baud1200,  
    SerialPortBaudRate_Baud2400,  
    SerialPortBaudRate_Baud4800,  
    SerialPortBaudRate_Baud9600,  
    SerialPortBaudRate_Baud14400,  
    SerialPortBaudRate_Baud19200,  
    SerialPortBaudRate_Baud38400,  
    SerialPortBaudRate_Baud57600,  
    SerialPortBaudRate_Baud115200,  
    SerialPortBaudRate_Baud230400,  
    SerialPortBaudRate_Baud460800,  
    SerialPortBaudRate_Baud921600,  
    NUM_SERIALPORTBAUDRATE }  
  
• enum SerialPortParityEnums {  
    SerialPortParity_None,  
    SerialPortParity_Odd,  
    SerialPortParity_Even,  
    SerialPortParity_Mark,  
    SerialPortParity_Space,  
    NUM_SERIALPORTPARITY }  
  
• enum SerialPortSelectorEnums {  
    SerialPortSelector_SerialPort0,  
    NUM_SERIALPORTSELECTOR }  
  
• enum SerialPortStopBitsEnums {  
    SerialPortStopBits_Bits1,  
    SerialPortStopBits_Bits1AndAHalf,  
    SerialPortStopBits_Bits2,  
    NUM_SERIALPORTSTOPBITS }
```

- enum `SerialPortSourceEnums` {  
    `SerialPortSource_Line0`,  
    `SerialPortSource_Line1`,  
    `SerialPortSource_Line2`,  
    `SerialPortSource_Line3`,  
    `SerialPortSource_Off`,  
    `NUM_SERIALPORTSOURCE` }
- enum `SequencerModeEnums` {  
    `SequencerMode_Off`,  
    `SequencerMode_On`,  
    `NUM_SEQUENCERMODE` }
- enum `SequencerConfigurationValidEnums` {  
    `SequencerConfigurationValid_No`,  
    `SequencerConfigurationValid_Yes`,  
    `NUM_SEQUENCERCONFIGURATIONVALID` }
- enum `SequencerSetValidEnums` {  
    `SequencerSetValid_No`,  
    `SequencerSetValid_Yes`,  
    `NUM_SEQUENCERSETVALID` }
- enum `SequencerTriggerActivationEnums` {  
    `SequencerTriggerActivation_RisingEdge`,  
    `SequencerTriggerActivation_FallingEdge`,  
    `SequencerTriggerActivation_AnyEdge`,  
    `SequencerTriggerActivation_LevelHigh`,  
    `SequencerTriggerActivation_LevelLow`,  
    `NUM_SEQUENCERTRIGGERACTIVATION` }
- enum `SequencerConfigurationModeEnums` {  
    `SequencerConfigurationMode_Off`,  
    `SequencerConfigurationMode_On`,  
    `NUM_SEQUENCERCONFIGURATIONMODE` }
- enum `SequencerTriggerSourceEnums` {  
    `SequencerTriggerSource_Off`,  
    `SequencerTriggerSource_FrameStart`,  
    `NUM_SEQUENCERTRIGGERSOURCE` }
- enum `TransferQueueModeEnums` {  
    `TransferQueueMode_FirstInFirstOut`,  
    `NUM_TRANSFERQUEUEMODE` }
- enum `TransferOperationModeEnums` {  
    `TransferOperationMode_Continuous`,  
    `TransferOperationMode_MultiBlock`,  
    `NUM_TRANSFEROPERATIONMODE` }
- enum `TransferControlModeEnums` {  
    `TransferControlMode_Basic`,  
    `TransferControlMode_Automatic`,  
    `TransferControlMode_UserControlled`,  
    `NUM_TRANSFERCONTROLMODE` }
- enum `ChunkGainSelectorEnums` {  
    `ChunkGainSelector_All`,  
    `ChunkGainSelector_Red`,  
    `ChunkGainSelector_Green`,  
    `ChunkGainSelector_Blue`,  
    `NUM_CHUNKGAINSELECTOR` }
- enum `ChunkSelectorEnums` {

- ChunkSelector\_Image,
  - ChunkSelector\_CRC,
  - ChunkSelector\_FrameID,
  - ChunkSelector\_OffsetX,
  - ChunkSelector\_OffsetY,
  - ChunkSelector\_Width,
  - ChunkSelector\_Height,
  - ChunkSelector\_ExposureTime,
  - ChunkSelector\_Gain,
  - ChunkSelector\_BlackLevel,
  - ChunkSelector\_PixelFormat,
  - ChunkSelector\_Timestamp,
  - ChunkSelector\_SequencerSetActive,
  - ChunkSelector\_SerialData,
  - ChunkSelector\_ExposureEndLineStatusAll,
  - NUM\_CHUNKSELECTOR }
- enum ChunkBlackLevelSelectorEnums {
  - ChunkBlackLevelSelector\_All,
  - NUM\_CHUNKBLACKLEVELSELECTOR }
- enum ChunkPixelFormatEnums {
  - ChunkPixelFormat\_Mono8,
  - ChunkPixelFormat\_Mono12Packed,
  - ChunkPixelFormat\_Mono16,
  - ChunkPixelFormat\_RGB8Packed,
  - ChunkPixelFormat\_YUV422Packed,
  - ChunkPixelFormat\_BayerGR8,
  - ChunkPixelFormat\_BayerRG8,
  - ChunkPixelFormat\_BayerGB8,
  - ChunkPixelFormat\_BayerBG8,
  - ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY,
  - NUM\_CHUNKPIXELFORMAT }
- enum FileOperationStatusEnums {
  - FileOperationStatus\_Success,
  - FileOperationStatus\_Failure,
  - FileOperationStatus\_Overflow,
  - NUM\_FILEOPERATIONSTATUS }
- enum FileOpenModeEnums {
  - FileOpenMode\_Read,
  - FileOpenMode\_Write,
  - FileOpenMode\_ReadWrite,
  - NUM\_FILEOPENMODE }
- enum FileOperationSelectorEnums {
  - FileOperationSelector\_Open,
  - FileOperationSelector\_Close,
  - FileOperationSelector\_Read,
  - FileOperationSelector\_Write,
  - FileOperationSelector\_Delete,
  - NUM\_FILEOPERATIONSELECTOR }
- enum FileSelectorEnums {
  - FileSelector\_UserSetDefault,
  - FileSelector\_UserSet0,
  - FileSelector\_UserSet1,
  - FileSelector\_UserFile1,
  - FileSelector\_SerialPort0,
  - NUM\_FILESELECTOR }
- enum BinningSelectorEnums {

```
BinningSelector_All,  
BinningSelector_Sensor,  
BinningSelector_ISP,  
NUM_BINNINGSELECTOR }  
• enum TestPatternGeneratorSelectorEnums {  
    TestPatternGeneratorSelector_Sensor,  
    TestPatternGeneratorSelector_PipelineStart,  
    NUM_TESTPATTERNGENERATORSELECTOR }  
• enum TestPatternEnums {  
    TestPattern_Off,  
    TestPattern_Increment,  
    TestPattern_SensorTestPattern,  
    NUM_TESTPATTERN }  
• enum PixelColorFilterEnums {  
    PixelColorFilter_None,  
    PixelColorFilter_BayerRG,  
    PixelColorFilter_BayerGB,  
    PixelColorFilter_BayerGR,  
    PixelColorFilter_BayerBG,  
    NUM_PIXELCOLORFILTER }  
• enum AdcBitDepthEnums {  
    AdcBitDepth_Bit8,  
    AdcBitDepth_Bit10,  
    AdcBitDepth_Bit12,  
    AdcBitDepth_Bit14,  
    NUM_ADCBITDEPTH }  
• enum DecimationHorizontalModeEnums {  
    DecimationHorizontalMode_Discard,  
    NUM_DECIMATIONHORIZONTALMODE }  
• enum BinningVerticalModeEnums {  
    BinningVerticalMode_Sum,  
    BinningVerticalMode_Average,  
    NUM_BINNINGVERTICALMODE }  
• enum PixelSizeEnums {  
    PixelSize_Bpp1,  
    PixelSize_Bpp2,  
    PixelSize_Bpp4,  
    PixelSize_Bpp8,  
    PixelSize_Bpp10,  
    PixelSize_Bpp12,  
    PixelSize_Bpp14,  
    PixelSize_Bpp16,  
    PixelSize_Bpp20,  
    PixelSize_Bpp24,  
    PixelSize_Bpp30,  
    PixelSize_Bpp32,  
    PixelSize_Bpp36,  
    PixelSize_Bpp48,  
    PixelSize_Bpp64,  
    PixelSize_Bpp96,  
    NUM_PIXELSIZE }  
• enum DecimationSelectorEnums {  
    DecimationSelector_All,  
    DecimationSelector_Sensor,  
    NUM_DECIMATIONSELECTOR }  
• enum ImageCompressionModeEnums {  
    ImageCompressionMode_Off,  
    ImageCompressionMode_Lossless,
```

NUM\_IMAGECOMPRESSIONMODE }

- enum BinningHorizontalModeEnums {  
    BinningHorizontalMode\_Sum,  
    BinningHorizontalMode\_Average,  
    NUM\_BINNINGHORIZONTALMODE }

- enum PixelFormatEnums {

PixelFormat\_Mono8,  
PixelFormat\_Mono16,  
PixelFormat\_RGB8Packed,  
PixelFormat\_BayerGR8,  
PixelFormat\_BayerRG8,  
PixelFormat\_BayerGB8,  
PixelFormat\_BayerBG8,  
PixelFormat\_BayerGR16,  
PixelFormat\_BayerRG16,  
PixelFormat\_BayerGB16,  
PixelFormat\_BayerBG16,  
PixelFormat\_Mono12Packed,  
PixelFormat\_BayerGR12Packed,  
PixelFormat\_BayerRG12Packed,  
PixelFormat\_BayerGB12Packed,  
PixelFormat\_BayerBG12Packed,  
PixelFormat\_YUV411Packed,  
PixelFormat\_YUV422Packed,  
PixelFormat\_YUV444Packed,  
PixelFormat\_Mono12p,  
PixelFormat\_BayerGR12p,  
PixelFormat\_BayerRG12p,  
PixelFormat\_BayerGB12p,  
PixelFormat\_BayerBG12p,  
PixelFormat\_YCbCr8,  
PixelFormat\_YCbCr422\_8,  
PixelFormat\_YCbCr411\_8,  
PixelFormat\_BGR8,  
PixelFormat\_BGRa8,  
PixelFormat\_Mono10Packed,  
PixelFormat\_BayerGR10Packed,  
PixelFormat\_BayerRG10Packed,  
PixelFormat\_BayerGB10Packed,  
PixelFormat\_BayerBG10Packed,  
PixelFormat\_Mono10p,  
PixelFormat\_BayerGR10p,  
PixelFormat\_BayerRG10p,  
PixelFormat\_BayerGB10p,  
PixelFormat\_BayerBG10p,  
PixelFormat\_Mono1p,  
PixelFormat\_Mono2p,  
PixelFormat\_Mono4p,  
PixelFormat\_Mono8s,  
PixelFormat\_Mono10,  
PixelFormat\_Mono12,  
PixelFormat\_Mono14,  
PixelFormat\_BayerBG10,  
PixelFormat\_BayerBG12,  
PixelFormat\_BayerGB10,  
PixelFormat\_BayerGB12,  
PixelFormat\_BayerGR10,  
PixelFormat\_BayerGR12,  
PixelFormat\_BayerRG10,  
PixelFormat\_BayerRG12,  
PixelFormat\_RGBa8,  
PixelFormat\_RGBa10,  
PixelFormat\_RGBa10p,  
PixelFormat\_RGBa12,  
PixelFormat\_RGBa12p,  
PixelFormat\_RGBa14,  
PixelFormat\_RGBa16,  
PixelFormat\_RGB8,  
PixelFormat\_RGB8\_Planar,  
PixelFormat\_RGB10,  
PixelFormat\_RGB10\_Planar,

- NUM\_PIXELFORMAT }
- enum DecimationVerticalModeEnums {  
DecimationVerticalMode\_Discard,  
NUM\_DECIMATIONVERTICALMODE }
- enum LineModeEnums {  
LineMode\_Input,  
LineMode\_Output,  
NUM\_LINEMODE }
- enum LineSourceEnums {  
LineSource\_Off,  
LineSource\_Line0,  
LineSource\_Line1,  
LineSource\_Line2,  
LineSource\_Line3,  
LineSource\_UserOutput0,  
LineSource\_UserOutput1,  
LineSource\_UserOutput2,  
LineSource\_UserOutput3,  
LineSource\_Counter0Active,  
LineSource\_Counter1Active,  
LineSource\_LogicBlock0,  
LineSource\_LogicBlock1,  
LineSource\_ExposureActive,  
LineSource\_FrameTriggerWait,  
LineSource\_SerialPort0,  
LineSource\_PPSSignal,  
LineSource\_AllPixel,  
LineSource\_AnyPixel,  
NUM\_LINESOURCE }
- enum LineInputFilterSelectorEnums {  
LineInputFilterSelector\_Deglintch,  
LineInputFilterSelector\_Debounce,  
NUM\_LINEINPUTFILTERSELECTOR }
- enum UserOutputSelectorEnums {  
UserOutputSelector\_UserOutput0,  
UserOutputSelector\_UserOutput1,  
UserOutputSelector\_UserOutput2,  
UserOutputSelector\_UserOutput3,  
NUM\_USEROUTPUTSELECTOR }
- enum LineFormatEnums {  
LineFormat\_NoConnect,  
LineFormat\_TriState,  
LineFormat\_TTL,  
LineFormat\_LVDS,  
LineFormat\_RS422,  
LineFormat\_OptoCoupled,  
LineFormat\_OpenDrain,  
NUM\_LINEFORMAT }
- enum LineSelectorEnums {  
LineSelector\_Line0,  
LineSelector\_Line1,  
LineSelector\_Line2,  
LineSelector\_Line3,  
NUM\_LINESELECTOR }
- enum ExposureActiveModeEnums {  
ExposureActiveMode\_Line1,  
ExposureActiveMode\_AnyPixels,  
ExposureActiveMode\_AllPixels,



- NUM\_EXPOSUREACTIVEMODE }
- enum CounterTriggerActivationEnums {  
CounterTriggerActivation\_LevelLow,  
CounterTriggerActivation\_LevelHigh,  
CounterTriggerActivation\_FallingEdge,  
CounterTriggerActivation\_RisingEdge,  
CounterTriggerActivation\_AnyEdge,  
NUM\_COUNTERTRIGGERACTIVATION }
- enum CounterSelectorEnums {  
CounterSelector\_Counter0,  
CounterSelector\_Counter1,  
NUM\_COUNTERSELECTOR }
- enum CounterStatusEnums {  
CounterStatus\_CounterIdle,  
CounterStatus\_CounterTriggerWait,  
CounterStatus\_CounterActive,  
CounterStatus\_CounterCompleted,  
CounterStatus\_CounterOverflow,  
NUM\_COUNTERSTATUS }
- enum CounterTriggerSourceEnums {  
CounterTriggerSource\_Off,  
CounterTriggerSource\_Line0,  
CounterTriggerSource\_Line1,  
CounterTriggerSource\_Line2,  
CounterTriggerSource\_Line3,  
CounterTriggerSource\_UserOutput0,  
CounterTriggerSource\_UserOutput1,  
CounterTriggerSource\_UserOutput2,  
CounterTriggerSource\_UserOutput3,  
CounterTriggerSource\_Counter0Start,  
CounterTriggerSource\_Counter1Start,  
CounterTriggerSource\_Counter0End,  
CounterTriggerSource\_Counter1End,  
CounterTriggerSource\_LogicBlock0,  
CounterTriggerSource\_LogicBlock1,  
CounterTriggerSource\_ExposureStart,  
CounterTriggerSource\_ExposureEnd,  
CounterTriggerSource\_FrameTriggerWait,  
NUM\_COUNTERTRIGGERSOURCE }
- enum CounterResetSourceEnums {  
CounterResetSource\_Off,  
CounterResetSource\_Line0,  
CounterResetSource\_Line1,  
CounterResetSource\_Line2,  
CounterResetSource\_Line3,  
CounterResetSource\_UserOutput0,  
CounterResetSource\_UserOutput1,  
CounterResetSource\_UserOutput2,  
CounterResetSource\_UserOutput3,  
CounterResetSource\_Counter0Start,  
CounterResetSource\_Counter1Start,  
CounterResetSource\_Counter0End,  
CounterResetSource\_Counter1End,  
CounterResetSource\_LogicBlock0,  
CounterResetSource\_LogicBlock1,  
CounterResetSource\_ExposureStart,  
CounterResetSource\_ExposureEnd,  
CounterResetSource\_FrameTriggerWait,

```

    NUM_COUNTERRESETSOURCE }
    • enum CounterEventSourceEnums {
        CounterEventSource_Off,
        CounterEventSource_MHzTick,
        CounterEventSource_Line0,
        CounterEventSource_Line1,
        CounterEventSource_Line2,
        CounterEventSource_Line3,
        CounterEventSource_UserOutput0,
        CounterEventSource_UserOutput1,
        CounterEventSource_UserOutput2,
        CounterEventSource_UserOutput3,
        CounterEventSource_Counter0Start,
        CounterEventSource_Counter1Start,
        CounterEventSource_Counter0End,
        CounterEventSource_Counter1End,
        CounterEventSource_LogicBlock0,
        CounterEventSource_LogicBlock1,
        CounterEventSource_ExposureStart,
        CounterEventSource_ExposureEnd,
        CounterEventSource_FrameTriggerWait,
        NUM_COUNTEREVENTSOURCE }
    • enum CounterEventActivationEnums {
        CounterEventActivation_LevelLow,
        CounterEventActivation_LevelHigh,
        CounterEventActivation_FallingEdge,
        CounterEventActivation_RisingEdge,
        CounterEventActivation_AnyEdge,
        NUM_COUNTEREVENTACTIVATION }
    • enum CounterResetActivationEnums {
        CounterResetActivation_LevelLow,
        CounterResetActivation_LevelHigh,
        CounterResetActivation_FallingEdge,
        CounterResetActivation_RisingEdge,
        CounterResetActivation_AnyEdge,
        NUM_COUNTERRESETACTIVATION }
    • enum DeviceTypeEnums {
        DeviceType_Transmitter,
        DeviceType_Receiver,
        DeviceType_Transceiver,
        DeviceType_Peripheral,
        NUM_DEVICETYPE }
    • enum DeviceConnectionStatusEnums {
        DeviceConnectionStatus_Active,
        DeviceConnectionStatus_Inactive,
        NUM_DEVICECONNECTIONSTATUS }
    • enum DeviceLinkThroughputLimitModeEnums {
        DeviceLinkThroughputLimitMode_On,
        DeviceLinkThroughputLimitMode_Off,
        NUM_DEVICELINKTHROUGHPUTLIMITMODE }
    • enum DeviceLinkHeartbeatModeEnums {
        DeviceLinkHeartbeatMode_On,
        DeviceLinkHeartbeatMode_Off,
        NUM_DEVICELINKHEARTBEATMODE }
    • enum DeviceStreamChannelTypeEnums {
        DeviceStreamChannelType_Transmitter,
        DeviceStreamChannelType_Receiver,
        NUM_DEVICESTREAMCHANNELTYPE }

```

- enum [DeviceStreamChannelEndiannessEnums](#) {  
    [DeviceStreamChannelEndianness\\_Big](#),  
    [DeviceStreamChannelEndianness\\_Little](#),  
    [NUM\\_DEVICESTREAMCHANNELENDIANNESS](#) }
- enum [DeviceClockSelectorEnums](#) {  
    [DeviceClockSelector\\_Sensor](#),  
    [DeviceClockSelector\\_SensorDigitization](#),  
    [DeviceClockSelector\\_CameraLink](#),  
    [NUM\\_DEVICECLOCKSELECTOR](#) }
- enum [DeviceSerialPortSelectorEnums](#) {  
    [DeviceSerialPortSelector\\_CameraLink](#),  
    [NUM\\_DEVICSERIALPORTSELECTOR](#) }
- enum [DeviceSerialPortBaudRateEnums](#) {  
    [DeviceSerialPortBaudRate\\_Baud9600](#),  
    [DeviceSerialPortBaudRate\\_Baud19200](#),  
    [DeviceSerialPortBaudRate\\_Baud38400](#),  
    [DeviceSerialPortBaudRate\\_Baud57600](#),  
    [DeviceSerialPortBaudRate\\_Baud115200](#),  
    [DeviceSerialPortBaudRate\\_Baud230400](#),  
    [DeviceSerialPortBaudRate\\_Baud460800](#),  
    [DeviceSerialPortBaudRate\\_Baud921600](#),  
    [NUM\\_DEVICSERIALPORTBAUDRATE](#) }
- enum [SensorTapsEnums](#) {  
    [SensorTaps\\_One](#),  
    [SensorTaps\\_Two](#),  
    [SensorTaps\\_Three](#),  
    [SensorTaps\\_Four](#),  
    [SensorTaps\\_Eight](#),  
    [SensorTaps\\_Ten](#),  
    [NUM\\_SENSORTAPS](#) }
- enum [SensorDigitizationTapsEnums](#) {  
    [SensorDigitizationTaps\\_One](#),  
    [SensorDigitizationTaps\\_Two](#),  
    [SensorDigitizationTaps\\_Three](#),  
    [SensorDigitizationTaps\\_Four](#),  
    [SensorDigitizationTaps\\_Eight](#),  
    [SensorDigitizationTaps\\_Ten](#),  
    [NUM\\_SENSORDIGITIZATIONTAPS](#) }
- enum [RegionSelectorEnums](#) {  
    [RegionSelector\\_Region0](#),  
    [RegionSelector\\_Region1](#),  
    [RegionSelector\\_Region2](#),  
    [RegionSelector\\_All](#),  
    [NUM\\_REGIONSELECTOR](#) }
- enum [RegionModeEnums](#) {  
    [RegionMode\\_Off](#),  
    [RegionMode\\_On](#),  
    [NUM\\_REGIONMODE](#) }
- enum [RegionDestinationEnums](#) {  
    [RegionDestination\\_Stream0](#),  
    [RegionDestination\\_Stream1](#),  
    [RegionDestination\\_Stream2](#),  
    [NUM\\_REGIONDESTINATION](#) }
- enum [ImageComponentSelectorEnums](#) {

```
ImageComponentSelector_Intensity,  
ImageComponentSelector_Color,  
ImageComponentSelector_Infrared,  
ImageComponentSelector_Ultraviolet,  
ImageComponentSelector_Range,  
ImageComponentSelector_Disparity,  
ImageComponentSelector_Confidence,  
ImageComponentSelector_Scatter,  
NUM_IMAGECOMPONENTSELECTOR }
```

- enum [PixelFormatInfoSelectorEnums](#) {

[PixelFormatInfoSelector\\_Mono1p,](#)  
[PixelFormatInfoSelector\\_Mono2p,](#)  
[PixelFormatInfoSelector\\_Mono4p,](#)  
[PixelFormatInfoSelector\\_Mono8,](#)  
[PixelFormatInfoSelector\\_Mono8s,](#)  
[PixelFormatInfoSelector\\_Mono10,](#)  
[PixelFormatInfoSelector\\_Mono10p,](#)  
[PixelFormatInfoSelector\\_Mono12,](#)  
[PixelFormatInfoSelector\\_Mono12p,](#)  
[PixelFormatInfoSelector\\_Mono14,](#)  
[PixelFormatInfoSelector\\_Mono16,](#)  
[PixelFormatInfoSelector\\_BayerBG8,](#)  
[PixelFormatInfoSelector\\_BayerBG10,](#)  
[PixelFormatInfoSelector\\_BayerBG10p,](#)  
[PixelFormatInfoSelector\\_BayerBG12,](#)  
[PixelFormatInfoSelector\\_BayerBG12p,](#)  
[PixelFormatInfoSelector\\_BayerBG16,](#)  
[PixelFormatInfoSelector\\_BayerGB8,](#)  
[PixelFormatInfoSelector\\_BayerGB10,](#)  
[PixelFormatInfoSelector\\_BayerGB10p,](#)  
[PixelFormatInfoSelector\\_BayerGB12,](#)  
[PixelFormatInfoSelector\\_BayerGB12p,](#)  
[PixelFormatInfoSelector\\_BayerGB16,](#)  
[PixelFormatInfoSelector\\_BayerGR8,](#)  
[PixelFormatInfoSelector\\_BayerGR10,](#)  
[PixelFormatInfoSelector\\_BayerGR10p,](#)  
[PixelFormatInfoSelector\\_BayerGR12,](#)  
[PixelFormatInfoSelector\\_BayerGR12p,](#)  
[PixelFormatInfoSelector\\_BayerGR16,](#)  
[PixelFormatInfoSelector\\_BayerRG8,](#)  
[PixelFormatInfoSelector\\_BayerRG10,](#)  
[PixelFormatInfoSelector\\_BayerRG10p,](#)  
[PixelFormatInfoSelector\\_BayerRG12,](#)  
[PixelFormatInfoSelector\\_BayerRG12p,](#)  
[PixelFormatInfoSelector\\_BayerRG16,](#)  
[PixelFormatInfoSelector\\_RGBa8,](#)  
[PixelFormatInfoSelector\\_RGBa10,](#)  
[PixelFormatInfoSelector\\_RGBa10p,](#)  
[PixelFormatInfoSelector\\_RGBa12,](#)  
[PixelFormatInfoSelector\\_RGBa12p,](#)  
[PixelFormatInfoSelector\\_RGBa14,](#)  
[PixelFormatInfoSelector\\_RGBa16,](#)  
[PixelFormatInfoSelector\\_RGB8,](#)  
[PixelFormatInfoSelector\\_RGB8\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB10,](#)  
[PixelFormatInfoSelector\\_RGB10\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB10p,](#)  
[PixelFormatInfoSelector\\_RGB10p32,](#)  
[PixelFormatInfoSelector\\_RGB12,](#)  
[PixelFormatInfoSelector\\_RGB12\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB12p,](#)  
[PixelFormatInfoSelector\\_RGB14,](#)  
[PixelFormatInfoSelector\\_RGB16,](#)  
[PixelFormatInfoSelector\\_RGB16\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB565p,](#)  
[PixelFormatInfoSelector\\_BGRa8,](#)  
[PixelFormatInfoSelector\\_BGRa10,](#)  
[PixelFormatInfoSelector\\_BGRa10p,](#)  
[PixelFormatInfoSelector\\_BGRa12,](#)  
[PixelFormatInfoSelector\\_BGRa12p,](#)  
[PixelFormatInfoSelector\\_BGRa14,](#)  
[PixelFormatInfoSelector\\_BGRa16,](#)  
[PixelFormatInfoSelector\\_BGR8,](#)  
[PixelFormatInfoSelector\\_BGR10,](#)  
[PixelFormatInfoSelector\\_BGR10p,](#)

- NUM\_PIXELFORMATINFOSELECTOR }
- enum DeinterlacingEnums {  
Deinterlacing\_Off,  
Deinterlacing\_LineDuplication,  
Deinterlacing\_Weave,  
NUM\_DEINTERLACING }
- enum ImageCompressionRateOptionEnums {  
ImageCompressionRateOption\_FixBitrate,  
ImageCompressionRateOption\_FixQuality,  
NUM\_IMAGECOMPRESSIONRATEOPTION }
- enum ImageCompressionJPEGFormatOptionEnums {  
ImageCompressionJPEGFormatOption\_Lossless,  
ImageCompressionJPEGFormatOption\_BaselineStandard,  
ImageCompressionJPEGFormatOption\_BaselineOptimized,  
ImageCompressionJPEGFormatOption\_Progressive,  
NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION }
- enum AcquisitionStatusSelectorEnums {  
AcquisitionStatusSelector\_AcquisitionTriggerWait,  
AcquisitionStatusSelector\_AcquisitionActive,  
AcquisitionStatusSelector\_AcquisitionTransfer,  
AcquisitionStatusSelector\_FrameTriggerWait,  
AcquisitionStatusSelector\_FrameActive,  
AcquisitionStatusSelector\_ExposureActive,  
NUM\_ACQUISITIONSTATUSSELECTION }
- enum ExposureTimeModeEnums {  
ExposureTimeMode\_Common,  
ExposureTimeMode\_Individual,  
NUM\_EXPOSURETIMEMODE }
- enum ExposureTimeSelectorEnums {  
ExposureTimeSelector\_Common,  
ExposureTimeSelector\_Red,  
ExposureTimeSelector\_Green,  
ExposureTimeSelector\_Blue,  
ExposureTimeSelector\_Cyan,  
ExposureTimeSelector\_Magenta,  
ExposureTimeSelector\_Yellow,  
ExposureTimeSelector\_Infrared,  
ExposureTimeSelector\_Ultraviolet,  
ExposureTimeSelector\_Stage1,  
ExposureTimeSelector\_Stage2,  
NUM\_EXPOSURETIMESELECTION }
- enum GainAutoBalanceEnums {  
GainAutoBalance\_Off,  
GainAutoBalance\_Once,  
GainAutoBalance\_Continuous,  
NUM\_GAINAUTOBALANCE }
- enum BlackLevelAutoEnums {  
BlackLevelAuto\_Off,  
BlackLevelAuto\_Once,  
BlackLevelAuto\_Continuous,  
NUM\_BLACKLEVELAUTO }
- enum BlackLevelAutoBalanceEnums {  
BlackLevelAutoBalance\_Off,  
BlackLevelAutoBalance\_Once,  
BlackLevelAutoBalance\_Continuous,  
NUM\_BLACKLEVELAUTOBALANCE }
- enum WhiteClipSelectorEnums {

```
WhiteClipSelector_All,  
WhiteClipSelector_Red,  
WhiteClipSelector_Green,  
WhiteClipSelector_Blue,  
WhiteClipSelector_Y,  
WhiteClipSelector_U,  
WhiteClipSelector_V,  
WhiteClipSelector_Tap1,  
WhiteClipSelector_Tap2,  
NUM_WHITECLIPSELECTOR }
```

- enum `TimerSelectorEnums` {  
    `TimerSelector_Timer0`,  
    `TimerSelector_Timer1`,  
    `TimerSelector_Timer2`,  
    `NUM_TIMERSELECTOR` }

- enum `TimerStatusEnums` {  
    `TimerStatus_TimerIdle`,  
    `TimerStatus_TimerTriggerWait`,  
    `TimerStatus_TimerActive`,  
    `TimerStatus_TimerCompleted`,  
    `NUM_TIMERSTATUS` }

- enum `TimerTriggerSourceEnums` {

```

TimerTriggerSource_Off,
TimerTriggerSource_AcquisitionTrigger,
TimerTriggerSource_AcquisitionStart,
TimerTriggerSource_AcquisitionEnd,
TimerTriggerSource_FrameTrigger,
TimerTriggerSource_FrameStart,
TimerTriggerSource_FrameEnd,
TimerTriggerSource_FrameBurstStart,
TimerTriggerSource_FrameBurstEnd,
TimerTriggerSource_LineTrigger,
TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

• enum TimerTriggerActivationEnums {
    TimerTriggerActivation_RisingEdge,
    TimerTriggerActivation_FallingEdge,
    TimerTriggerActivation_AnyEdge,
    TimerTriggerActivation_LevelHigh,
    TimerTriggerActivation_LevelLow,
    NUM_TIMERTRIGGERACTIVATION }

• enum EncoderSelectorEnums {
    EncoderSelector_Encoder0,
    EncoderSelector_Encoder1,
    EncoderSelector_Encoder2,
    NUM_ENCODERSELECTOR }

• enum EncoderSourceAEnums {

```



```
EncoderSourceA_Off,  
EncoderSourceA_Line0,  
EncoderSourceA_Line1,  
EncoderSourceA_Line2,  
NUM_ENCODERSOURCEA }
```

- `enum EncoderSourceBEnums {`  
EncoderSourceB\_Off,  
EncoderSourceB\_Line0,  
EncoderSourceB\_Line1,  
EncoderSourceB\_Line2,  
NUM\_ENCODERSOURCEB }

- `enum EncoderModeEnums {`  
EncoderMode\_FourPhase,  
EncoderMode\_HighResolution,  
NUM\_ENCODERMODE }

- `enum EncoderOutputModeEnums {`  
EncoderOutputMode\_Off,  
EncoderOutputMode\_PositionUp,  
EncoderOutputMode\_PositionDown,  
EncoderOutputMode\_DirectionUp,  
EncoderOutputMode\_DirectionDown,  
EncoderOutputMode\_Motion,  
NUM\_ENCODEROUTPUTMODE }

- `enum EncoderStatusEnums {`  
EncoderStatus\_EncoderUp,  
EncoderStatus\_EncoderDown,  
EncoderStatus\_EncoderIdle,  
EncoderStatus\_EncoderStatic,  
NUM\_ENCODERSTATUS }

- `enum EncoderResetSourceEnums {`

```

EncoderResetSource_Off,
EncoderResetSource_AcquisitionTrigger,
EncoderResetSource_AcquisitionStart,
EncoderResetSource_AcquisitionEnd,
EncoderResetSource_FrameTrigger,
EncoderResetSource_FrameStart,
EncoderResetSource_FrameEnd,
EncoderResetSource_ExposureStart,
EncoderResetSource_ExposureEnd,
EncoderResetSource_Line0,
EncoderResetSource_Line1,
EncoderResetSource_Line2,
EncoderResetSource_Counter0Start,
EncoderResetSource_Counter1Start,
EncoderResetSource_Counter2Start,
EncoderResetSource_Counter0End,
EncoderResetSource_Counter1End,
EncoderResetSource_Counter2End,
EncoderResetSource_Timer0Start,
EncoderResetSource_Timer1Start,
EncoderResetSource_Timer2Start,
EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,
EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }

• enum EncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }

• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }

• enum ActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }

• enum SourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,

```

```
NUM_SOURCESELECTOR }  
• enum TransferSelectorEnums {  
    TransferSelector_Stream0,  
    TransferSelector_Stream1,  
    TransferSelector_Stream2,  
    TransferSelector_All,  
    NUM_TRANSFERSELECTION }  
• enum TransferTriggerSelectorEnums {  
    TransferTriggerSelector_TransferStart,  
    TransferTriggerSelector_TransferStop,  
    TransferTriggerSelector_TransferAbort,  
    TransferTriggerSelector_TransferPause,  
    TransferTriggerSelector_TransferResume,  
    TransferTriggerSelector_TransferActive,  
    TransferTriggerSelector_TransferBurstStart,  
    TransferTriggerSelector_TransferBurstStop,  
    NUM_TRANSFERTRIGGERSELECTION }  
• enum TransferTriggerModeEnums {  
    TransferTriggerMode_Off,  
    TransferTriggerMode_On,  
    NUM_TRANSFERTRIGGERMODE }  
• enum TransferTriggerSourceEnums {  
    TransferTriggerSource_Line0,  
    TransferTriggerSource_Line1,  
    TransferTriggerSource_Line2,  
    TransferTriggerSource_Counter0Start,  
    TransferTriggerSource_Counter1Start,  
    TransferTriggerSource_Counter2Start,  
    TransferTriggerSource_Counter0End,  
    TransferTriggerSource_Counter1End,  
    TransferTriggerSource_Counter2End,  
    TransferTriggerSource_Timer0Start,  
    TransferTriggerSource_Timer1Start,  
    TransferTriggerSource_Timer2Start,  
    TransferTriggerSource_Timer0End,  
    TransferTriggerSource_Timer1End,  
    TransferTriggerSource_Timer2End,  
    TransferTriggerSource_SoftwareSignal0,  
    TransferTriggerSource_SoftwareSignal1,  
    TransferTriggerSource_SoftwareSignal2,  
    TransferTriggerSource_Action0,  
    TransferTriggerSource_Action1,  
    TransferTriggerSource_Action2,  
    NUM_TRANSFERTRIGGERSOURCE }  
• enum TransferTriggerActivationEnums {  
    TransferTriggerActivation_RisingEdge,  
    TransferTriggerActivation_FallingEdge,  
    TransferTriggerActivation_AnyEdge,  
    TransferTriggerActivation_LevelHigh,  
    TransferTriggerActivation_LevelLow,  
    NUM_TRANSFERTRIGGERACTIVATION }  
• enum TransferStatusSelectorEnums {  
    TransferStatusSelector_Streaming,  
    TransferStatusSelector_Paused,  
    TransferStatusSelector_Stopping,  
    TransferStatusSelector_Stopped,  
    TransferStatusSelector_QueueOverflow,  
    NUM_TRANSFERSTATUSSELECTION }
```

- enum [TransferComponentSelectorEnums](#) {  
[TransferComponentSelector\\_Red](#),  
[TransferComponentSelector\\_Green](#),  
[TransferComponentSelector\\_Blue](#),  
[TransferComponentSelector\\_All](#),  
[NUM\\_TRANSFERCOMPONENTSELECTOR](#) }
- enum [Scan3dDistanceUnitEnums](#) {  
[Scan3dDistanceUnit\\_Millimeter](#),  
[Scan3dDistanceUnit\\_Inch](#),  
[NUM\\_SCAN3DDISTANCEUNIT](#) }
- enum [Scan3dCoordinateSystemEnums](#) {  
[Scan3dCoordinateSystem\\_Cartesian](#),  
[Scan3dCoordinateSystem\\_Spherical](#),  
[Scan3dCoordinateSystem\\_Cylindrical](#),  
[NUM\\_SCAN3DCOORDINATESYSTEM](#) }
- enum [Scan3dOutputModeEnums](#) {  
[Scan3dOutputMode\\_UncalibratedC](#),  
[Scan3dOutputMode\\_CalibratedABC\\_Grid](#),  
[Scan3dOutputMode\\_CalibratedABC\\_PointCloud](#),  
[Scan3dOutputMode\\_CalibratedAC](#),  
[Scan3dOutputMode\\_CalibratedAC\\_Linescan](#),  
[Scan3dOutputMode\\_CalibratedC](#),  
[Scan3dOutputMode\\_CalibratedC\\_Linescan](#),  
[Scan3dOutputMode\\_RectifiedC](#),  
[Scan3dOutputMode\\_RectifiedC\\_Linescan](#),  
[Scan3dOutputMode\\_DisparityC](#),  
[Scan3dOutputMode\\_DisparityC\\_Linescan](#),  
[NUM\\_SCAN3DOUTPUTMODE](#) }
- enum [Scan3dCoordinateSystemReferenceEnums](#) {  
[Scan3dCoordinateSystemReference\\_Anchor](#),  
[Scan3dCoordinateSystemReference\\_Transformed](#),  
[NUM\\_SCAN3DCOORDINATESYSTEMREFERENCE](#) }
- enum [Scan3dCoordinateSelectorEnums](#) {  
[Scan3dCoordinateSelector\\_CoordinateA](#),  
[Scan3dCoordinateSelector\\_CoordinateB](#),  
[Scan3dCoordinateSelector\\_CoordinateC](#),  
[NUM\\_SCAN3DCOORDINATESELECTOR](#) }
- enum [Scan3dCoordinateTransformSelectorEnums](#) {  
[Scan3dCoordinateTransformSelector\\_RotationX](#),  
[Scan3dCoordinateTransformSelector\\_RotationY](#),  
[Scan3dCoordinateTransformSelector\\_RotationZ](#),  
[Scan3dCoordinateTransformSelector\\_TranslationX](#),  
[Scan3dCoordinateTransformSelector\\_TranslationY](#),  
[Scan3dCoordinateTransformSelector\\_TranslationZ](#),  
[NUM\\_SCAN3DCOORDINATETRANSFORMSELECTOR](#) }
- enum [Scan3dCoordinateReferenceSelectorEnums](#) {  
[Scan3dCoordinateReferenceSelector\\_RotationX](#),  
[Scan3dCoordinateReferenceSelector\\_RotationY](#),  
[Scan3dCoordinateReferenceSelector\\_RotationZ](#),  
[Scan3dCoordinateReferenceSelector\\_TranslationX](#),  
[Scan3dCoordinateReferenceSelector\\_TranslationY](#),  
[Scan3dCoordinateReferenceSelector\\_TranslationZ](#),  
[NUM\\_SCAN3DCOORDINATEREFERENCESELECTOR](#) }
- enum [ChunkImageComponentEnums](#) {

```
ChunkImageComponent_Intensity,  
ChunkImageComponent_Color,  
ChunkImageComponent_Infrared,  
ChunkImageComponent_Ultraviolet,  
ChunkImageComponent_Range,  
ChunkImageComponent_Disparity,  
ChunkImageComponent_Confidence,  
ChunkImageComponent_Scatter,  
NUM_CHUNKIMAGECOMPONENT }  
• enum ChunkCounterSelectorEnums {  
    ChunkCounterSelector_Counter0,  
    ChunkCounterSelector_Counter1,  
    ChunkCounterSelector_Counter2,  
    NUM_CHUNKCOUNTERSELECTOR }  
• enum ChunkTimerSelectorEnums {  
    ChunkTimerSelector_Timer0,  
    ChunkTimerSelector_Timer1,  
    ChunkTimerSelector_Timer2,  
    NUM_CHUNKTIMERSELECTOR }  
• enum ChunkEncoderSelectorEnums {  
    ChunkEncoderSelector_Encoder0,  
    ChunkEncoderSelector_Encoder1,  
    ChunkEncoderSelector_Encoder2,  
    NUM_CHUNKENCODERSELECTOR }  
• enum ChunkEncoderStatusEnums {  
    ChunkEncoderStatus_EncoderUp,  
    ChunkEncoderStatus_EncoderDown,  
    ChunkEncoderStatus_EncoderIdle,  
    ChunkEncoderStatus_EncoderStatic,  
    NUM_CHUNKENCODERSTATUS }  
• enum ChunkExposureTimeSelectorEnums {  
    ChunkExposureTimeSelector_Common,  
    ChunkExposureTimeSelector_Red,  
    ChunkExposureTimeSelector_Green,  
    ChunkExposureTimeSelector_Blue,  
    ChunkExposureTimeSelector_Cyan,  
    ChunkExposureTimeSelector_Magenta,  
    ChunkExposureTimeSelector_Yellow,  
    ChunkExposureTimeSelector_Infrared,  
    ChunkExposureTimeSelector_Ultraviolet,  
    ChunkExposureTimeSelector_Stage1,  
    ChunkExposureTimeSelector_Stage2,  
    NUM_CHUNKEXPOSURETIMESELECTOR }  
• enum ChunkSourceIDEnums {  
    ChunkSourceID_Source0,  
    ChunkSourceID_Source1,  
    ChunkSourceID_Source2,  
    NUM_CHUNKSOURCEID }  
• enum ChunkRegionIDEnums {  
    ChunkRegionID_Region0,  
    ChunkRegionID_Region1,  
    ChunkRegionID_Region2,  
    NUM_CHUNKREGIONID }  
• enum ChunkTransferStreamIDEnums {  
    ChunkTransferStreamID_Stream0,  
    ChunkTransferStreamID_Stream1,  
    ChunkTransferStreamID_Stream2,  
    ChunkTransferStreamID_Stream3,
```

NUM\_CHUNKTRANSFERSTREAMID }

- enum ChunkScan3dDistanceUnitEnums {  
 ChunkScan3dDistanceUnit\_Millimeter,  
 ChunkScan3dDistanceUnit\_Inch,  
 NUM\_CHUNKSCAN3DDISTANCEUNIT }
- enum ChunkScan3dOutputModeEnums {  
 ChunkScan3dOutputMode\_UncalibratedC,  
 ChunkScan3dOutputMode\_CalibratedABC\_Grid,  
 ChunkScan3dOutputMode\_CalibratedABC\_PointCloud,  
 ChunkScan3dOutputMode\_CalibratedAC,  
 ChunkScan3dOutputMode\_CalibratedAC\_Linescan,  
 ChunkScan3dOutputMode\_CalibratedC,  
 ChunkScan3dOutputMode\_CalibratedC\_Linescan,  
 ChunkScan3dOutputMode\_RectifiedC,  
 ChunkScan3dOutputMode\_RectifiedC\_Linescan,  
 ChunkScan3dOutputMode\_DisparityC,  
 ChunkScan3dOutputMode\_DisparityC\_Linescan,  
 NUM\_CHUNKSCAN3DOUTPUTMODE }
- enum ChunkScan3dCoordinateSystemEnums {  
 ChunkScan3dCoordinateSystem\_Cartesian,  
 ChunkScan3dCoordinateSystem\_Spherical,  
 ChunkScan3dCoordinateSystem\_Cylindrical,  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEM }
- enum ChunkScan3dCoordinateSystemReferenceEnums {  
 ChunkScan3dCoordinateSystemReference\_Anchor,  
 ChunkScan3dCoordinateSystemReference\_Transformed,  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }
- enum ChunkScan3dCoordinateSelectorEnums {  
 ChunkScan3dCoordinateSelector\_CoordinateA,  
 ChunkScan3dCoordinateSelector\_CoordinateB,  
 ChunkScan3dCoordinateSelector\_CoordinateC,  
 NUM\_CHUNKSCAN3DCOORDINATESELECTOR }
- enum ChunkScan3dCoordinateTransformSelectorEnums {  
 ChunkScan3dCoordinateTransformSelector\_RotationX,  
 ChunkScan3dCoordinateTransformSelector\_RotationY,  
 ChunkScan3dCoordinateTransformSelector\_RotationZ,  
 ChunkScan3dCoordinateTransformSelector\_TranslationX,  
 ChunkScan3dCoordinateTransformSelector\_TranslationY,  
 ChunkScan3dCoordinateTransformSelector\_TranslationZ,  
 NUM\_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }
- enum ChunkScan3dCoordinateReferenceSelectorEnums {  
 ChunkScan3dCoordinateReferenceSelector\_RotationX,  
 ChunkScan3dCoordinateReferenceSelector\_RotationY,  
 ChunkScan3dCoordinateReferenceSelector\_RotationZ,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationX,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationY,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationZ,  
 NUM\_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }
- enum DeviceTapGeometryEnums {

```

DeviceTapGeometry_Geometry_1X_1Y,
DeviceTapGeometry_Geometry_1X2_1Y,
DeviceTapGeometry_Geometry_1X2_1Y2,
DeviceTapGeometry_Geometry_2X_1Y,
DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
DeviceTapGeometry_Geometry_2XE_1Y2,
DeviceTapGeometry_Geometry_2XM_1Y,
DeviceTapGeometry_Geometry_2XM_1Y2,
DeviceTapGeometry_Geometry_1X_1Y2,
DeviceTapGeometry_Geometry_1X_2YE,
DeviceTapGeometry_Geometry_1X3_1Y,
DeviceTapGeometry_Geometry_3X_1Y,
DeviceTapGeometry_Geometry_1X,
DeviceTapGeometry_Geometry_1X2,
DeviceTapGeometry_Geometry_2X,
DeviceTapGeometry_Geometry_2XE,
DeviceTapGeometry_Geometry_2XM,
DeviceTapGeometry_Geometry_1X3,
DeviceTapGeometry_Geometry_3X,
DeviceTapGeometry_Geometry_1X4_1Y,
DeviceTapGeometry_Geometry_4X_1Y,
DeviceTapGeometry_Geometry_2X2_1Y,
DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
DeviceTapGeometry_Geometry_1X2_2YE,
DeviceTapGeometry_Geometry_2X_2YE,
DeviceTapGeometry_Geometry_2XE_2YE,
DeviceTapGeometry_Geometry_2XM_2YE,
DeviceTapGeometry_Geometry_1X4,
DeviceTapGeometry_Geometry_4X,
DeviceTapGeometry_Geometry_2X2,
DeviceTapGeometry_Geometry_2X2E,
DeviceTapGeometry_Geometry_2X2M,
DeviceTapGeometry_Geometry_1X8_1Y,
DeviceTapGeometry_Geometry_8X_1Y,
DeviceTapGeometry_Geometry_4X2_1Y,
DeviceTapGeometry_Geometry_2X2E_2YE,
DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

```

- enum `GevPhysicalLinkConfigurationEnums` {  
`GevPhysicalLinkConfiguration_SingleLink`,  
`GevPhysicalLinkConfiguration_MultiLink`,  
`GevPhysicalLinkConfiguration_StaticLAG`,  
`GevPhysicalLinkConfiguration_DynamicLAG`,  
`NUM_GEVPHYSICALLINKCONFIGURATION` }
- enum `GevCurrentPhysicalLinkConfigurationEnums` {  
`GevCurrentPhysicalLinkConfiguration_SingleLink`,  
`GevCurrentPhysicalLinkConfiguration_MultiLink`,  
`GevCurrentPhysicalLinkConfiguration_StaticLAG`,  
`GevCurrentPhysicalLinkConfiguration_DynamicLAG`,  
`NUM_GEVCURRENTPHYSICALLINKCONFIGURATION` }

- enum [GevIPConfigurationStatusEnums](#) {  
    [GevIPConfigurationStatus\\_None](#),  
    [GevIPConfigurationStatus\\_PersistentIP](#),  
    [GevIPConfigurationStatus\\_DHCP](#),  
    [GevIPConfigurationStatus\\_LLA](#),  
    [GevIPConfigurationStatus\\_ForceIP](#),  
    [NUM\\_GEVIPCONFIGURATIONSTATUS](#) }
  
- enum [GevGVCPExtendedStatusCodesSelectorEnums](#) {  
    [GevGVCPExtendedStatusCodesSelector\\_Version1\\_1](#),  
    [GevGVCPExtendedStatusCodesSelector\\_Version2\\_0](#),  
    [NUM\\_GEVGVCPEXTENDEDSTATUSCODESSELECTOR](#) }
  
- enum [GevGVSPExtendedIDModeEnums](#) {  
    [GevGVSPExtendedIDMode\\_Off](#),  
    [GevGVSPExtendedIDMode\\_On](#),  
    [NUM\\_GEVGVSPEXTENDEDIDMODE](#) }
  
- enum [CIConfigurationEnums](#) {  
    [CIConfiguration\\_Base](#),  
    [CIConfiguration\\_Medium](#),  
    [CIConfiguration\\_Full](#),  
    [CIConfiguration\\_DualBase](#),  
    [CIConfiguration\\_EightyBit](#),  
    [NUM\\_CLCONFIGURATION](#) }
  
- enum [CITimeSlotsCountEnums](#) {  
    [CITimeSlotsCount\\_One](#),  
    [CITimeSlotsCount\\_Two](#),  
    [CITimeSlotsCount\\_Three](#),  
    [NUM\\_CLTIMESLOTSCOUNT](#) }
  
- enum [CxpLinkConfigurationStatusEnums](#) {



```
CxpLinkConfigurationStatus_None,  
CxpLinkConfigurationStatus_Pending,  
CxpLinkConfigurationStatus_CXP1_X1,  
CxpLinkConfigurationStatus_CXP2_X1,  
CxpLinkConfigurationStatus_CXP3_X1,  
CxpLinkConfigurationStatus_CXP5_X1,  
CxpLinkConfigurationStatus_CXP6_X1,  
CxpLinkConfigurationStatus_CXP1_X2,  
CxpLinkConfigurationStatus_CXP2_X2,  
CxpLinkConfigurationStatus_CXP3_X2,  
CxpLinkConfigurationStatus_CXP5_X2,  
CxpLinkConfigurationStatus_CXP6_X2,  
CxpLinkConfigurationStatus_CXP1_X3,  
CxpLinkConfigurationStatus_CXP2_X3,  
CxpLinkConfigurationStatus_CXP3_X3,  
CxpLinkConfigurationStatus_CXP5_X3,  
CxpLinkConfigurationStatus_CXP6_X3,  
CxpLinkConfigurationStatus_CXP1_X4,  
CxpLinkConfigurationStatus_CXP2_X4,  
CxpLinkConfigurationStatus_CXP3_X4,  
CxpLinkConfigurationStatus_CXP5_X4,  
CxpLinkConfigurationStatus_CXP6_X4,  
CxpLinkConfigurationStatus_CXP1_X5,  
CxpLinkConfigurationStatus_CXP2_X5,  
CxpLinkConfigurationStatus_CXP3_X5,  
CxpLinkConfigurationStatus_CXP5_X5,  
CxpLinkConfigurationStatus_CXP6_X5,  
CxpLinkConfigurationStatus_CXP1_X6,  
CxpLinkConfigurationStatus_CXP2_X6,  
CxpLinkConfigurationStatus_CXP3_X6,  
CxpLinkConfigurationStatus_CXP5_X6,  
CxpLinkConfigurationStatus_CXP6_X6,  
NUM_CXPLINKCONFIGURATIONSTATUS }
```

• enum [CxpLinkConfigurationPreferredEnums](#) {

```
CxpLinkConfigurationPreferred_CXP1_X1,  
CxpLinkConfigurationPreferred_CXP2_X1,  
CxpLinkConfigurationPreferred_CXP3_X1,  
CxpLinkConfigurationPreferred_CXP5_X1,  
CxpLinkConfigurationPreferred_CXP6_X1,  
CxpLinkConfigurationPreferred_CXP1_X2,  
CxpLinkConfigurationPreferred_CXP2_X2,  
CxpLinkConfigurationPreferred_CXP3_X2,  
CxpLinkConfigurationPreferred_CXP5_X2,  
CxpLinkConfigurationPreferred_CXP6_X2,  
CxpLinkConfigurationPreferred_CXP1_X3,  
CxpLinkConfigurationPreferred_CXP2_X3,  
CxpLinkConfigurationPreferred_CXP3_X3,  
CxpLinkConfigurationPreferred_CXP5_X3,  
CxpLinkConfigurationPreferred_CXP6_X3,  
CxpLinkConfigurationPreferred_CXP1_X4,  
CxpLinkConfigurationPreferred_CXP2_X4,  
CxpLinkConfigurationPreferred_CXP3_X4,  
CxpLinkConfigurationPreferred_CXP5_X4,  
CxpLinkConfigurationPreferred_CXP6_X4,  
CxpLinkConfigurationPreferred_CXP1_X5,  
CxpLinkConfigurationPreferred_CXP2_X5,  
CxpLinkConfigurationPreferred_CXP3_X5,  
CxpLinkConfigurationPreferred_CXP5_X5,  
CxpLinkConfigurationPreferred_CXP6_X5,  
CxpLinkConfigurationPreferred_CXP1_X6,  
CxpLinkConfigurationPreferred_CXP2_X6,  
CxpLinkConfigurationPreferred_CXP3_X6,  
CxpLinkConfigurationPreferred_CXP5_X6,  
CxpLinkConfigurationPreferred_CXP6_X6,  
NUM_CXPLINKCONFIGURATIONPREFERRED }
```

- enum [CxpLinkConfigurationEnums](#) {

```
CxpLinkConfiguration_Auto,  
CxpLinkConfiguration_CXP1_X1,  
CxpLinkConfiguration_CXP2_X1,  
CxpLinkConfiguration_CXP3_X1,  
CxpLinkConfiguration_CXP5_X1,  
CxpLinkConfiguration_CXP6_X1,  
CxpLinkConfiguration_CXP1_X2,  
CxpLinkConfiguration_CXP2_X2,  
CxpLinkConfiguration_CXP3_X2,  
CxpLinkConfiguration_CXP5_X2,  
CxpLinkConfiguration_CXP6_X2,  
CxpLinkConfiguration_CXP1_X3,  
CxpLinkConfiguration_CXP2_X3,  
CxpLinkConfiguration_CXP3_X3,  
CxpLinkConfiguration_CXP5_X3,  
CxpLinkConfiguration_CXP6_X3,  
CxpLinkConfiguration_CXP1_X4,  
CxpLinkConfiguration_CXP2_X4,  
CxpLinkConfiguration_CXP3_X4,  
CxpLinkConfiguration_CXP5_X4,  
CxpLinkConfiguration_CXP6_X4,  
CxpLinkConfiguration_CXP1_X5,  
CxpLinkConfiguration_CXP2_X5,  
CxpLinkConfiguration_CXP3_X5,  
CxpLinkConfiguration_CXP5_X5,  
CxpLinkConfiguration_CXP6_X5,  
CxpLinkConfiguration_CXP1_X6,  
CxpLinkConfiguration_CXP2_X6,  
CxpLinkConfiguration_CXP3_X6,  
CxpLinkConfiguration_CXP5_X6,  
CxpLinkConfiguration_CXP6_X6,  
NUM_CXPLINKCONFIGURATION }
```

- enum [CxpConnectionTestModeEnums](#) {  
    [CxpConnectionTestMode\\_Off](#),  
    [CxpConnectionTestMode\\_Mode1](#),  
    [NUM\\_CXPCONNECTIONTESTMODE](#) }

- enum [CxpPoCxpStatusEnums](#) {  
    [CxpPoCxpStatus\\_Auto](#),  
    [CxpPoCxpStatus\\_Off](#),  
    [CxpPoCxpStatus\\_Tripped](#),  
    [NUM\\_CXPPOCXPSTATUS](#) }

- enum [Error](#) {

```

SPINNAKER_ERR_SUCCESS = 0,
SPINNAKER_ERR_ERROR = -1001,
SPINNAKER_ERR_NOT_INITIALIZED = -1002,
SPINNAKER_ERR_NOT_IMPLEMENTED = -1003,
SPINNAKER_ERR_RESOURCE_IN_USE = -1004,
SPINNAKER_ERR_ACCESS_DENIED = -1005,
SPINNAKER_ERR_INVALID_HANDLE = -1006,
SPINNAKER_ERR_INVALID_ID = -1007,
SPINNAKER_ERR_NO_DATA = -1008,
SPINNAKER_ERR_INVALID_PARAMETER = -1009,
SPINNAKER_ERR_IO = -1010,
SPINNAKER_ERR_TIMEOUT = -1011,
SPINNAKER_ERR_ABORT = -1012,
SPINNAKER_ERR_INVALID_BUFFER = -1013,
SPINNAKER_ERR_NOT_AVAILABLE = -1014,
SPINNAKER_ERR_INVALID_ADDRESS = -1015,
SPINNAKER_ERR_BUFFER_TOO_SMALL = -1016,
SPINNAKER_ERR_INVALID_INDEX = -1017,
SPINNAKER_ERR_PARSING_CHUNK_DATA = -1018,
SPINNAKER_ERR_INVALID_VALUE = -1019,
SPINNAKER_ERR_RESOURCE_EXHAUSTED = -1020,
SPINNAKER_ERR_OUT_OF_MEMORY = -1021,
SPINNAKER_ERR_BUSY = -1022,
GENICAM_ERR_INVALID_ARGUMENT = -2001,
GENICAM_ERR_OUT_OF_RANGE = -2002,
GENICAM_ERR_PROPERTY = -2003,
GENICAM_ERR_RUN_TIME = -2004,
GENICAM_ERR_LOGICAL = -2005,
GENICAM_ERR_ACCESS = -2006,
GENICAM_ERR_TIMEOUT = -2007,
GENICAM_ERR_DYNAMIC_CAST = -2008,
GENICAM_ERR_GENERIC = -2009,
GENICAM_ERR_BAD_ALLOCATION = -2010,
SPINNAKER_ERR_IM_CONVERT = -3001,
SPINNAKER_ERR_IM_COPY = -3002,
SPINNAKER_ERR_IM_MALLOC = -3003,
SPINNAKER_ERR_IM_NOT_SUPPORTED = -3004,
SPINNAKER_ERR_IM_HISTOGRAM_RANGE = -3005,
SPINNAKER_ERR_IM_HISTOGRAM_MEAN = -3006,
SPINNAKER_ERR_IM_MIN_MAX = -3007,
SPINNAKER_ERR_IM_COLOR_CONVERSION = -3008,
SPINNAKER_ERR_IM_DECOMPRESSION = -3009,
SPINNAKER_ERR_CUSTOM_ID = -10000 }

```

*Spinnaker enum definitions.*

- enum EventType {
 

```

SPINNAKER_EVENT_ARRIVAL_REMOVAL,
SPINNAKER_EVENT_DEVICE,
SPINNAKER_EVENT_DEVICE_SPECIFIC,
SPINNAKER_EVENT_NEW_BUFFER,
SPINNAKER_EVENT_LOGGING_EVENT,
SPINNAKER_EVENT_UNKNOWN }

```

*Event types in Spinnaker.*

- enum PixelFormatNamespaceID {
 

```

SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,
SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,
SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,

```

```
SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }
```

*This enum represents the namespace in which the TL specific pixel format resides.*

- enum `ColorProcessingAlgorithm` {  
`DEFAULT`,  
`NO_COLOR_PROCESSING`,  
`NEAREST_NEIGHBOR`,  
`EDGE_SENSING`,  
`HQ_LINEAR`,  
`RIGOROUS`,  
`IPP`,  
`DIRECTIONAL_FILTER`,  
`WEIGHTED_DIRECTIONAL_FILTER` }

*Color processing algorithms.*

- enum `PolarizationAlgorithm` {  
`NO_POLARIZATION`,  
`QUADRANT_I0_GRAYSCALE`,  
`QUADRANT_I45_GRAYSCALE`,  
`QUADRANT_I90_GRAYSCALE`,  
`QUADRANT_I135_GRAYSCALE`,  
`STOKES_S0_GRAYSCALE`,  
`STOKES_S0_HEATMAP`,  
`STOKES_S1_GRAYSCALE`,  
`STOKES_S1_HEATMAP`,  
`STOKES_S2_GRAYSCALE`,  
`STOKES_S2_HEATMAP`,  
`DOLP_GRAYSCALE`,  
`DOLP_HEATMAP`,  
`AOP_GRAYSCALE`,  
`AOP_HEATMAP` }
- enum `PolarizationResolution` {  
`QUARTER_RESOLUTION`,  
`FULL_RESOLUTION` }
- enum `HeatMapColor` {  
`HEATMAP_BLACK` = 1,  
`HEATMAP_BLUE` = 2,  
`HEATMAP_CYAN` = 3,  
`HEATMAP_GREEN` = 4,  
`HEATMAP_YELLOW` = 5,  
`HEATMAP_RED` = 6,  
`HEATMAP_WHITE` = 7 }
- enum `ImageFileFormat` {  
`FROM_FILE_EXT` = -1,  
`PGM`,  
`PPM`,  
`BMP`,  
`JPEG`,  
`JPEG2000`,  
`TIFF`,  
`PNG`,  
`RAW`,  
`JPEG12_C`,  
`IMAGE_FILE_FORMAT_FORCE_32BITS` = 0x7FFFFFFF }

*File formats to be used for saving images to disk.*

- enum `ImageStatus` {

```

IMAGE_UNKNOWN_ERROR = -1,
IMAGE_NO_ERROR = 0,
IMAGE_CRC_CHECK_FAILED = 1,
IMAGE_DATA_OVERFLOW = 2,
IMAGE_MISSING_PACKETS = 3,
IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT = 4,
IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT = 5,
IMAGE_PACKETID_INCONSISTENT = 6,
IMAGE_MISSING_LEADER = 7,
IMAGE_MISSING_TRAILER = 8,
IMAGE_DATA_INCOMPLETE = 9,
IMAGE_INFO_INCONSISTENT = 10,
IMAGE_CHUNK_DATA_INVALID = 11,
IMAGE_NO_SYSTEM_RESOURCES = 12 }

```

*Status of images returned from GetNextImage() call.*

- enum `StatisticsChannel` {  
`GREY`,  
`RED`,  
`GREEN`,  
`BLUE`,  
`HUE`,  
`SATURATION`,  
`LIGHTNESS`,  
`NUM_STATISTICS_CHANNELS` }

*Channels that allow statistics to be calculated.*

- enum `SpinnakerLogLevel` {  
`LOG_LEVEL_OFF` = -1,  
`LOG_LEVEL_FATAL` = 0,  
`LOG_LEVEL_ALERT` = 100,  
`LOG_LEVEL_CRIT` = 200,  
`LOG_LEVEL_ERROR` = 300,  
`LOG_LEVEL_WARN` = 400,  
`LOG_LEVEL_NOTICE` = 500,  
`LOG_LEVEL_INFO` = 600,  
`LOG_LEVEL_DEBUG` = 700,  
`LOG_LEVEL_NOTSET` = 800 }

*log levels*

- enum `PayloadTypeInfoIds` {  
`PAYLOAD_TYPE_UNKNOWN` = 0,  
`PAYLOAD_TYPE_IMAGE` = 1,  
`PAYLOAD_TYPE_RAW_DATA` = 2,  
`PAYLOAD_TYPE_FILE` = 3,  
`PAYLOAD_TYPE_CHUNK_DATA` = 4,  
`PAYLOAD_TYPE_JPEG` = 5,  
`PAYLOAD_TYPE_JPEG2000` = 6,  
`PAYLOAD_TYPE_H264` = 7,  
`PAYLOAD_TYPE_CHUNK_ONLY` = 8,  
`PAYLOAD_TYPE_DEVICE_SPECIFIC` = 9,  
`PAYLOAD_TYPE_MULTI_PART` = 10,  
`PAYLOAD_TYPE_CUSTOM_ID` = 1000,  
`PAYLOAD_TYPE_EXTENDED_CHUNK` = 1001 }
- enum `ActionCommandStatus` {  
`ACTION_COMMAND_STATUS_OK` = 0,  
`ACTION_COMMAND_STATUS_NO_REF_TIME` = 0x8013,  
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,  
`ACTION_COMMAND_STATUS_ACTION_LATE` = 0x8016,  
`ACTION_COMMAND_STATUS_ERROR` = 0x8FFF }

*Possible Status Codes Returned from Action Command.*

- enum `PixelFormatIntType` {  
`IntType_UINT8`,  
`IntType_INT8`,  
`IntType_UINT10`,  
`IntType_UINT10p`,  
`IntType_UINT10P`,  
`IntType_UINT12`,  
`IntType_UINT12p`,  
`IntType_UINT12P`,  
`IntType_UINT14`,  
`IntType_UINT16`,  
`IntType_FLOAT32`,  
`IntType_UNKNOWN` }

*Possible integer types and packing used in a pixel format.*

- enum `StreamTypeEnum` {  
`StreamType_Mixed`,  
`StreamType_Custom`,  
`StreamType_GEV`,  
`StreamType_CL`,  
`StreamType_IIDC`,  
`StreamType_UVC`,  
`StreamType_CXP`,  
`StreamType_CLHS`,  
`StreamType_U3V`,  
`StreamType_ETHERNET`,  
`StreamType_PCI`,  
`NUMSTREAMTYPE` }

*The enum definitions for TL Device nodes from the transport layer .xml files.*

- enum `StreamDefaultBufferCountModeEnum` {  
`StreamDefaultBufferCountMode_Manual`,  
`StreamDefaultBufferCountMode_Auto`,  
`NUMSTREAMDEFAULTBUFFERCOUNTMODE` }
- enum `StreamBufferCountModeEnum` {  
`StreamBufferCountMode_Manual`,  
`StreamBufferCountMode_Auto`,  
`NUMSTREAMBUFFERCOUNTMODE` }
- enum `StreamBufferHandlingModeEnum` {  
`StreamBufferHandlingMode_OldestFirst`,  
`StreamBufferHandlingMode_OldestFirstOverwrite`,  
`StreamBufferHandlingMode_NewestFirst`,  
`StreamBufferHandlingMode_NewestFirstOverwrite`,  
`StreamBufferHandlingMode_NewestOnly`,  
`NUMSTREAMBUFFERHANDLINGMODE` }
- enum `DeviceTypeEnum` {  
`DeviceType_Mixed`,  
`DeviceType_Custom`,  
`DeviceType_GEV`,  
`DeviceType_CL`,  
`DeviceType_IIDC`,  
`DeviceType_UVC`,  
`DeviceType_CXP`,  
`DeviceType_CLHS`,  
`DeviceType_U3V`,  
`DeviceType_ETHERNET`,  
`DeviceType_PCI`,  
`NUMDEVICETYPE` }

- enum [DeviceAccessStatusEnum](#) {  
[DeviceAccessStatus\\_Unknown](#),  
[DeviceAccessStatus\\_ReadWrite](#),  
[DeviceAccessStatus\\_ReadOnly](#),  
[DeviceAccessStatus\\_NoAccess](#),  
[NUMDEVICEACCESSSTATUS](#) }
- enum [GevCCPEnum](#) {  
[GevCCP\\_EnumEntry\\_GevCCP\\_OpenAccess](#),  
[GevCCP\\_EnumEntry\\_GevCCP\\_ExclusiveAccess](#),  
[GevCCP\\_EnumEntry\\_GevCCP\\_ControlAccess](#),  
[NUMGEVCCP](#) }
- enum [GUIXMLLocationEnum](#) {  
[GUIXMLLocation\\_Device](#),  
[GUIXMLLocation\\_Host](#),  
[NUMGUIXMLLOCATION](#) }
- enum [GenICamXMLLocationEnum](#) {  
[GenICamXMLLocation\\_Device](#),  
[GenICamXMLLocation\\_Host](#),  
[NUMGENICAMXMLLOCATION](#) }
- enum [DeviceEndiannessMechanismEnum](#) {  
[DeviceEndiannessMechanism\\_Legacy](#),  
[DeviceEndiannessMechanism\\_Standard](#),  
[NUMDEVICEENDIANESSMECHANISM](#) }
- enum [DeviceCurrentSpeedEnum](#) {  
[DeviceCurrentSpeed\\_UnknownSpeed](#),  
[DeviceCurrentSpeed\\_LowSpeed](#),  
[DeviceCurrentSpeed\\_FullSpeed](#),  
[DeviceCurrentSpeed\\_HighSpeed](#),  
[DeviceCurrentSpeed\\_SuperSpeed](#),  
[NUMDEVICECURRENTSPEED](#) }
- enum [POEStatusEnum](#) {  
[POEStatus\\_NotSupported](#),  
[POEStatus\\_PowerOff](#),  
[POEStatus\\_PowerOn](#),  
[NUMPOESTATUS](#) }

## Functions

- class [DEPRECATED\\_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER\_API A↔  
 VIRecorder  
*Provides the functionality for the user to record images to an AVI file.*

## Variables

- const uint64\_t [EVENT\\_TIMEOUT\\_NONE](#) = 0  
*Timeout values for getting next image, device, or interface event.*
- const uint64\_t [EVENT\\_TIMEOUT\\_INFINITE](#) = 0xFFFFFFFFFFFFFFFF

## 9.2 Spinnaker::GenApi Namespace Reference

### Classes

- class [AutoLock](#)



- class [BooleanNode](#)  
*Interface for string properties.*
- class [CategoryNode](#)  
*Interface for string properties.*
- class [CChunkAdapter](#)  
*Connects a chunked buffer to a node map.*
- class [CChunkAdapterDcam](#)  
*Connects a chunked DCAM buffer to a node map.*
- class [CChunkAdapterGeneric](#)
- class [CChunkAdapterGEV](#)  
*Connects a chunked DCAM buffer to a node map.*
- class [CChunkAdapterU3V](#)  
*Connects a chunked U3V buffer to a node map.*
- class [CChunkPort](#)  
*Port attachable to a chunk in a buffer.*
- class [CEnumerationTRef](#)  
*Interface for string properties.*
- class [CEventAdapter](#)  
*Delivers Events to ports.*
- class [CEventAdapter1394](#)  
*Distribute the events to the node map.*
- class [CEventAdapterGeneric](#)  
*Connects a generic event to a node map.*
- class [CEventAdapterGEV](#)  
*Connects a GigE [Event](#) to a node map.*
- class [CEventAdapterU3V](#)  
*Connects a U3V [Event](#) to a node map.*
- class [CEventPort](#)  
*Port attachable to an event.*
- class [CFeatureBag](#)  
*Bag holding streamable features of a nodetree.*
- class [CFloatPtr](#)  
*SmartPointer for IFloat interface pointer.*
- class [CGeneric\\_XMLLoaderParams](#)  
*Empty base class used by class [CNodeMapRef](#) as generic template argument.*
- class [CLock](#)  
*A lock class.*
- class [CLockEx](#)  
*This class is for testing purposes only.*
- class [CNodeCallback](#)  
*callback body instance for INode pointers*
- class [CNodeMapFactory](#)  
*The node map factory is used for creating node maps from camera description files.*
- class [CNodeMapRef](#)  
*SmartPointer for NodeMaps with create function.*
- class [CNodeMapRefT](#)  
*SmartPointer template for NodeMaps with create function.*
- class [CommandNode](#)  
*Interface for string properties.*
- class [Counter](#)  
*Definition of a simple [Counter](#) class.*

- class [CPointer](#)  
*Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.*
- class [CPortImpl](#)  
*Standard implementation for a port.*
- class [CPortWriteList](#)  
*Container holding a list of port write commands.*
- class [CRegisterPortImpl](#)  
*Standard implementation for a port using a register based transport layer.*
- class [CSelectorSet](#)  
*The set of selectors selecting a given node.*
- class [CTestPortStruct](#)  
*Implements a register spaces based on a C++ struct.*
- class [double\\_autovector\\_t](#)  
*Vector of doubles with reference counting.*
- class [EAccessModeClass](#)  
*Holds conversion methods for the access mode enumeration.*
- class [ECachingModeClass](#)  
*Holds conversion methods for the caching mode enumeration.*
- class [EDisplayNotationClass](#)  
*Holds conversion methods for the notation type of floats.*
- class [EEndianessClass](#)  
*Holds conversion methods for the endianess enumeration.*
- class [EGenApiSchemaVersionClass](#)  
*helper class converting `EGenApiSchemaVersion` from and to string*
- class [EInputDirectionClass](#)  
*Holds conversion methods for the notation type of floats.*
- class [ENamespaceClass](#)  
*Holds conversion methods for the namespace enumeration.*
- class [EnumEntryNode](#)  
*[Interface](#) for string properties.*
- class [EnumNode](#)  
*[Interface](#) for string properties.*
- class [ERepresentationClass](#)  
*Holds conversion methods for the representation enumeration.*
- class [ESignClass](#)  
*Holds conversion methods for the sign enumeration.*
- class [ESlopeClass](#)  
*Holds conversion methods for the converter formulas.*
- class [EStandardNameSpaceClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [EVisibilityClass](#)  
*Holds conversion methods for the visibility enumeration.*
- class [EYesNoClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [FileProtocolAdapter](#)  
*Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.*
- class [FloatNode](#)  
*[Interface](#) for string properties.*
- class [FloatRegNode](#)  
*[Interface](#) for string properties.*
- class [Function\\_NodeCallback](#)

- Container for a function pointer.*
- class [IDevFileStreamBase](#)
- class [IDevFileStreamBuf](#)
- class [int64\\_autovector\\_t](#)
- Vector of integers with reference counting.*
- class [IntegerNode](#)
- Interface for string properties.*
- class [IntRegNode](#)
- Interface for string properties.*
- class [Member\\_NodeCallback](#)
- Container for a member function pointer.*
- class [Node](#)
- class common to all nodes*
- class [NodeMap](#)
- Smart pointer template for NodeMaps with create function.*
- class [ODevFileStreamBase](#)
- class [ODevFileStreamBuf](#)
- class [PortNode](#)
- Interface for value properties.*
- class [PortRecorder](#)
- Interface for recording write commands on a port.*
- class [PortReplay](#)
- Interface for replaying write commands on a port.*
- class [RegisterNode](#)
- Interface for string properties.*
- class [SpinTestCamera](#)
- class [StringNode](#)
- Interface for string properties.*
- class [StringRegNode](#)
- Interface for string properties.*
- class [ValueNode](#)
- Interface for value properties.*

## Typedefs

- typedef [BooleanNode](#) [CBooleanRef](#)
- typedef [CategoryNode](#) [CCategoryRef](#)
- typedef [CommandNode](#) [CCommandRef](#)
- typedef [EnumEntryNode](#) [CEnumEntryRef](#)
- typedef [EnumNode](#) [CEnumerationRef](#)
- typedef [ODevFileStreamBase](#)< char, std::char\_traits< char > > [ODevFileStream](#)
- typedef [IDevFileStreamBase](#)< char, std::char\_traits< char > > [IDevFileStream](#)
- typedef [FloatNode](#) [CFloatRef](#)
- typedef node\_vector [NodeList\\_t](#)
- a list of node references*
- typedef intptr\_t [CallbackHandleType](#)
- the callback handle for nodes*
- typedef [IntegerNode](#) [CIntegerRef](#)
- typedef [Node](#) [CNodeRef](#)
- typedef [Node](#) [CSelectorRef](#)
- typedef [NodeMap](#) [CNodeMapRef](#)

- typedef [CPointer](#)< [IBase](#) > [CBasePtr](#)  
*SmartPointer for IBase interface pointer.*
- typedef [CPointer](#)< [INode](#), [IBase](#) > [CNodePtr](#)  
*SmartPointer for INode interface pointer.*
- typedef [CPointer](#)< [IValue](#) > [CValuePtr](#)  
*SmartPointer for IValue interface pointer.*
- typedef [CPointer](#)< [ICategory](#) > [CCategoryPtr](#)  
*SmartPointer for ICategory interface pointer.*
- typedef [CPointer](#)< [IBoolean](#) > [CBooleanPtr](#)  
*SmartPointer for IBoolean interface pointer.*
- typedef [CPointer](#)< [IInteger](#) > [CIntegerPtr](#)  
*SmartPointer for IInteger interface pointer.*
- typedef [CPointer](#)< [IString](#) > [CStringPtr](#)  
*SmartPointer for IString interface pointer.*
- typedef [CPointer](#)< [IRegister](#) > [CRegisterPtr](#)  
*SmartPointer for IRegister interface pointer.*
- typedef [CPointer](#)< [IEnumeration](#) > [CEnumerationPtr](#)  
*SmartPointer for IEnumeration interface pointer.*
- typedef [CPointer](#)< [IEnumEntry](#) > [CEnumEntryPtr](#)  
*SmartPointer for IEnumEntry interface pointer.*
- typedef [CPointer](#)< [IPort](#) > [CPortPtr](#)  
*SmartPointer for IPort interface pointer.*
- typedef [CPointer](#)< [IPortReplay](#) > [CPortReplayPtr](#)  
*SmartPointer for IPortReplay interface pointer.*
- typedef [CPointer](#)< [IPortRecorder](#) > [CPortRecorderPtr](#)  
*SmartPointer for IPortRecorder interface pointer.*
- typedef [CPointer](#)< [IPortWriteList](#), [IPortWriteList](#) > [CPortWriteListPtr](#)  
*SmartPointer for IPortWriteList interface pointer.*
- typedef [CPointer](#)< [IChunkPort](#) > [CChunkPortPtr](#)  
*SmartPointer for IChunkPort interface pointer.*
- typedef [CPointer](#)< [INodeMap](#), [INodeMap](#) > [CNodeMapPtr](#)  
*SmartPointer for INodeMap interface pointer.*
- typedef [CPointer](#)< [INodeMapDyn](#), [INodeMap](#) > [CNodeMapDynPtr](#)  
*SmartPointer for INodeMapDyn interface pointer.*
- typedef [CPointer](#)< [IDeviceInfo](#), [INodeMap](#) > [CDeviceInfoPtr](#)  
*SmartPointer for IDeviceInfo interface pointer.*
- typedef [CPointer](#)< [ISelector](#) > [CSelectorPtr](#)  
*SmartPointer for ISelector interface pointer.*
- typedef [CPointer](#)< [ICommand](#) > [CCommandPtr](#)  
*SmartPointer for ICommand interface pointer.*
- typedef [CPointer](#)< [IPortConstruct](#) > [CPortConstructPtr](#)  
*SmartPointer for IPortConstruct interface pointer.*
- typedef [PortNode](#) [CPortRef](#)
- typedef [PortRecorder](#) [CPortRecorderRef](#)  
*Reference to an IPortRecorder pointer.*
- typedef [RegisterNode](#) [CRegisterRef](#)
- typedef [StringNode](#) [CStringRef](#)
- typedef [GenlCam::gcstring\\_vector](#) [StringList\\_t](#)  
*A list of strings.*
- typedef [ValueNode](#) [CValueRef](#)

## Enumerations

- enum [GVCP\\_MESSAGE\\_TAGS](#) {  
[TAG\\_EVENT\\_CMD](#) = 0xc0,  
[TAG\\_EVENTDATA\\_CMD](#) = 0xc2 }
- enum [ECallbackType](#) {  
[cbPostInsideLock](#) = 1,  
[cbPostOutsideLock](#) = 2 }  
*the type of callback*
- enum [ECacheUsage\\_t](#) {  
[CacheUsage\\_Automatic](#),  
[CacheUsage\\_ForceWrite](#),  
[CacheUsage\\_ForceRead](#),  
[CacheUsage\\_Ignore](#) }  
*Lists the cache usage strategies.*
- enum [EContentType\\_t](#) {  
[ContentType\\_Xml](#),  
[ContentType\\_ZippedXml](#) }  
*Lists the processable file types.*
- enum [ESign](#) {  
[Signed](#),  
[Unsigned](#),  
[\\_UndefinedSign](#) }  
*signed or unsigned integers*
- enum [EAccessMode](#) {  
[NI](#),  
[NA](#),  
[WO](#),  
[RO](#),  
[RW](#),  
[\\_UndefinedAccesMode](#),  
[\\_CycleDetectAccesMode](#) }  
*access mode of a node*
- enum [EVisibility](#) {  
[Beginner](#) = 0,  
[Expert](#) = 1,  
[Guru](#) = 2,  
[Invisible](#) = 3,  
[\\_UndefinedVisibility](#) = 99 }  
*recommended visibility of a node*
- enum [ECachingMode](#) {  
[NoCache](#),  
[WriteThrough](#),  
[WriteAround](#),  
[\\_UndefinedCachingMode](#) }  
*caching mode of a register*
- enum [ERepresentation](#) {  
[Linear](#),  
[Logarithmic](#),  
[Boolean](#),  
[PureNumber](#),  
[HexNumber](#),  
[IPV4Address](#),  
[MACAddress](#),  
[\\_UndefinedRepresentation](#) }  
*recommended representation of a node value*

- enum [EEndianness](#) {  
[BigEndian](#),  
[LittleEndian](#),  
[\\_UndefinedEndian](#) }  
*Endianness of a value in a register.*
- enum [ENameSpace](#) {  
[Custom](#),  
[Standard](#),  
[\\_UndefinedNameSpace](#) }  
*Defines if a node name is standard or custom.*
- enum [EStandardNameSpace](#) {  
[None](#),  
[GEV](#),  
[IIDC](#),  
[CL](#),  
[USB](#),  
[\\_UndefinedStandardNameSpace](#) }  
*Defines from which standard namespace a node name comes from.*
- enum [EYesNo](#) {  
[Yes](#) = 1,  
[No](#) = 0,  
[\\_UndefinedYesNo](#) = 2 }  
*Defines the choices of a Yes/No alternative.*
- enum [ESlope](#) {  
[Increasing](#),  
[Decreasing](#),  
[Varying](#),  
[Automatic](#),  
[\\_UndefinedESlope](#) }  
*typedef for formula type*
- enum [EXMLValidation](#) {  
[xvLoad](#) = 0x00000001L,  
[xvCycles](#) = 0x00000002L,  
[xvSFNC](#) = 0x00000004L,  
[xvDefault](#) = 0x00000000L,  
[xvAll](#) = 0xffffffffL,  
[\\_UndefinedEXMLValidation](#) = 0x80000000L }  
*typedef describing the different validity checks which can be performed on an XML file*
- enum [EDisplayNotation](#) {  
[fnAutomatic](#),  
[fnFixed](#),  
[fnScientific](#),  
[\\_UndefinedEDisplayNotation](#) }  
*typedef for float notation*
- enum [EInterfaceType](#) {  
[intfIValue](#),  
[intfIBase](#),  
[intfInteger](#),  
[intfBoolean](#),  
[intfCommand](#),  
[intfFloat](#),  
[intfString](#),  
[intfRegister](#),  
[intfCategory](#),  
[intfEnumeration](#),  
[intfEnumEntry](#),  
[intfIPort](#) }

- typedef for interface type*
- enum [ELinkType](#) {  
[ctParentNodes](#),  
[ctReadingChildren](#),  
[ctWritingChildren](#),  
[ctInvalidatingChildren](#),  
[ctDependingNodes](#),  
[ctTerminalNodes](#) }
- typedef for link type*
- enum [EIncMode](#) {  
[noIncrement](#),  
[fixedIncrement](#),  
[listIncrement](#) }
- typedef for increment mode*
- enum [EInputDirection](#) {  
[idFrom](#),  
[idTo](#),  
[idNone](#) }
- typedef for link type*
- enum [EGenApiSchemaVersion](#) {  
[v1\\_0](#) = 1,  
[v1\\_1](#) = 2,  
[\\_Undefined](#) = -1 }
- GenApi schema version.*

## Functions

- void [SPINNAKER\\_API SET\\_GUID](#) (SPIN\_GUID &name, uint32\_t l, uint16\_t w1, uint16\_t w2, uint8\_t b1, uint8\_t b2, uint8\_t b3, uint8\_t b4, uint8\_t b5, uint8\_t b6, uint8\_t b7, uint8\_t b8)
- virtual void [operator=](#) (bool Value)  
*Set node value.*
- virtual bool [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false) const =0  
*Get node value.*
- virtual bool [operator\(\)](#) () const  
*Get node value.*
- virtual [EYesNo](#) [CacheChunkData](#) () const =0  
*Indicates if the chunk a adapter must hold a cached version of the chunk data.*
- virtual bool [IsDone](#) (bool [Verify](#)=true)=0  
*Query whether the command is executed.*
- virtual [GenICam::gcstring](#) [GetVendorName](#) ()=0  
*Get the vendor name.*
- virtual [GenICam::gcstring](#) [GetToolTip](#) ()=0  
*Get tool tip.*
- virtual [GenICam::gcstring](#) [GetStandardNameSpace](#) ()=0  
*Get the standard name space.*
- virtual void [GetGenApiVersion](#) ([GenICam::Version\\_t](#) &Version, uint16\_t &Build)=0  
*Get the version of the DLL's [GenApi](#) implementation.*
- virtual void [GetSchemaVersion](#) ([GenICam::Version\\_t](#) &Version)=0  
*Get the schema version number.*
- virtual void [GetDeviceVersion](#) ([GenICam::Version\\_t](#) &Version)=0  
*Get the version of the device description file.*
- virtual [GenICam::gcstring](#) [GetProductGuid](#) ()=0

- Get the Guid describing the product.*

  - virtual [GenlCam::gcstring GetVersionGuid](#) ()=0
- Get the Guid describing the product version.*

  - virtual [GenlCam::gcstring GetSymbolic](#) () const =0
- Get symbolic enum value.*

  - virtual double [GetNumericValue](#) ()=0
- Get double number associated with the entry.*

  - virtual bool [IsSelfClearing](#) ()=0
- Indicates if the corresponding EnumEntry is self clearing.*

  - virtual void [GetEntries](#) ([NodeList\\_t](#) &Entries)=0
- Get list of entry nodes.*

  - virtual [IEnumeration](#) & [operator=](#) (const [GenlCam::gcstring](#) &ValueStr)=0
- Set string node value.*

  - virtual void [SetIntValue](#) (int64\_t Value, bool [Verify](#)=true)=0
- Set integer node value.*

  - virtual [GenlCam::gcstring operator\\*](#) ()=0
- Get string node value.*

  - virtual int64\_t [GetIntValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
- Get integer node value.*

  - virtual [IEnumEntry](#) \* [GetEntryByName](#) (const [GenlCam::gcstring](#) &Symbolic)=0
- Get an entry node by name.*

  - virtual [IEnumEntry](#) \* [GetEntry](#) (const int64\_t IntValue)=0
- Get an entry node by its IntValue.*

  - virtual [IEnumEntry](#) \* [GetCurrentEntry](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
- Get the current entry.*

  - virtual [IEnumeration](#) & [operator=](#) (EnumT Value)=0
- Set node value.*

  - virtual [IEnumEntry](#) \* [GetEntry](#) (const EnumT Value)=0
- returns the EnumEntry object belonging to the Value*

  - virtual [IFloat](#) & [operator=](#) (double Value)=0
- Set node value.*

  - virtual double [GetMin](#) ()=0
- Get minimum value allowed.*

  - virtual double [GetMax](#) ()=0
- Get maximum value allowed.*

  - virtual bool [HasInc](#) ()=0
- True if the float has a constant increment.*

  - virtual [EIncMode](#) [GetIncMode](#) ()=0
- Get increment mode.*

  - virtual double [GetInc](#) ()=0
- Get the constant increment if there is any.*

  - virtual [double\\_autovector\\_t](#) [GetListOfValidValues](#) (bool bounded=true)=0
- Get list of valid value.*

  - virtual [ERepresentation](#) [GetRepresentation](#) ()=0
- Get recommended representation.*

  - virtual [GenlCam::gcstring GetUnit](#) () const =0
- Get the physical unit name.*

  - virtual [EDisplayNotation](#) [GetDisplayNotation](#) () const =0
- Get the way the float should be converted to a string.*

  - virtual int64\_t [GetDisplayPrecision](#) () const =0
- Get the precision to be used when converting the float to a string.*



- virtual void [ImposeMin](#) (double Value)=0  
*Restrict minimum value.*
- virtual void [ImposeMax](#) (double Value)=0  
*Restrict maximum value.*
- virtual [Integer](#) & [operator=](#) (int64\_t Value)=0  
*Set node value.*
- virtual void [ImposeMin](#) (int64\_t Value)=0  
*Restrict minimum value.*
- virtual void [ImposeMax](#) (int64\_t Value)=0  
*Restrict maximum value.*
- virtual [GenApi::ENamespace](#) [GetNameSpace](#) () const =0  
*Get name space.*
- virtual [EVisibility](#) [GetVisibility](#) () const =0  
*Get the recommended visibility of the node.*
- virtual void [InvalidateNode](#) ()=0  
*Indicates that the node's value may have changed.*
- virtual bool [IsCacheable](#) () const =0  
*Is the node value cacheable.*
- virtual [EYesNo](#) [IsAccessModeCacheable](#) () const =0  
*True if the AccessMode can be cached.*
- virtual [ECachingMode](#) [GetCachingMode](#) () const =0  
*Get Caching Mode.*
- virtual int64\_t [GetPollingTime](#) () const =0  
*recommended polling time (for non-cacheable nodes)*
- virtual [GenICam::gcstring](#) [GetDescription](#) () const =0  
*Get a long description of the node.*
- virtual [GenICam::gcstring](#) [GetDisplayName](#) () const =0  
*Get a name string for display.*
- virtual [GenICam::gcstring](#) [GetDeviceName](#) () const =0  
*Get a name of the device.*
- virtual void [GetChildren](#) ([GenApi::NodeList\\_t](#) &Children, [ELinkType](#) LinkType=[ctReadingChildren](#)) const =0  
*Get all nodes this node directly depends on.*
- virtual void [GetParents](#) ([GenApi::NodeList\\_t](#) &Parents) const =0  
*Gets all nodes this node is directly depending on.*
- virtual [CallbackHandleType](#) [RegisterCallback](#) ([CNodeCallback](#) \*pCallback)=0  
*Register change callback Takes ownership of the CNodeCallback object.*
- virtual bool [DeregisterCallback](#) ([CallbackHandleType](#) hCallback)=0  
*De register change callback Destroys CNodeCallback object.*
- virtual [INodeMap](#) \* [GetNodeMap](#) () const =0  
*Retrieves the central node map.*
- virtual [GenICam::gcstring](#) [GetEventID](#) () const =0  
*Get the EventId of the node.*
- virtual bool [IsStreamable](#) () const =0  
*True if the node is streamable.*
- virtual void [GetPropertyNames](#) ([GenICam::gcstring\\_vector](#) &PropertyNames) const =0  
*Returns a list of the names all properties set during initialization.*
- virtual bool [GetProperty](#) (const [GenICam::gcstring](#) &PropertyName, [GenICam::gcstring](#) &ValueStr, [GenICam::gcstring](#) &AttributeStr)=0  
*Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.*
- virtual void [ImposeAccessMode](#) ([EAccessMode](#) ImposedAccessMode)=0

- Imposes an access mode to the natural access mode of the node.*

  - virtual void **ImposeVisibility** (**EVisibility** ImposedVisibility)=0

*Imposes a visibility to the natural visibility of the node.*
- virtual **INode** \* **GetAlias** () const =0

*Retrieves the a node which describes the same feature in a different way.*
- virtual **INode** \* **GetCastAlias** () const =0

*Retrieves the a node which describes the same feature so that it can be casted.*
- virtual **GenICam::gcstring** **GetDocuURL** () const =0

*Gets a URL pointing to the documentation of that feature.*
- virtual bool **IsDeprecated** () const =0

*True if the node should not be used any more.*
- virtual **EInterfaceType** **GetPrincipalInterfaceType** () const =0

*Get the type of the main interface of a node.*
- virtual bool **IsFeature** () const =0

*True if the node can be reached via category nodes from a category node named "Root".*
- virtual bool **operator==** (int nullPtr) const =0
- virtual bool **operator!=** (int nullPtr) const =0
- bool **IsReadable** (**EAccessMode** AccessMode)

*Tests if readable.*
- bool **IsReadable** (const **IBase** \*p)

*Checks if a node is readable.*
- bool **IsReadable** (const **IBase** &r)

*Checks if a node is readable.*
- bool **IsWritable** (**EAccessMode** AccessMode)

*Tests if writable.*
- bool **IsWritable** (const **IBase** \*p)

*Checks if a node is writable.*
- bool **IsWritable** (const **IBase** &r)

*Checks if a node is writable.*
- bool **IsImplemented** (**EAccessMode** AccessMode)

*Tests if implemented.*
- bool **IsImplemented** (const **IBase** \*p)

*Checks if a node is implemented.*
- bool **IsImplemented** (const **IBase** &r)

*Checks if a node is implemented.*
- bool **IsAvailable** (**EAccessMode** AccessMode)

*Tests if available.*
- bool **IsAvailable** (const **IBase** \*p)

*Checks if a node is available.*
- bool **IsAvailable** (const **IBase** &r)

*Checks if a node is available.*
- **EAccessMode** **Combine** (**EAccessMode** Peter, **EAccessMode** Paul)

*Computes which access mode the two guards allow together.*
- bool **IsVisible** (**EVisibility** Visibility, **EVisibility** MaxVisibility)

*Tests Visibility CAVE : this relies on the EVisibility enum's coding.*
- **EVisibility** **Combine** (**EVisibility** Peter, **EVisibility** Paul)

*Computes which visibility the two guards allow together.*
- bool **IsCacheable** (**ECachingMode** CachingMode)

*Tests Cacheability.*
- **ECachingMode** **Combine** (**ECachingMode** Peter, **ECachingMode** Paul)

*Computes which CachingMode results from a combination.*

- virtual [INode](#) \* [GetNode](#) (const [GenICam::gcstring](#) &Name) const =0  
*Retrieves the node from the central map by Name.*
- virtual void [InvalidateNodes](#) () const =0  
*Invalidates all nodes.*
- virtual bool [Connect](#) ([IPort](#) \*pPort, const [GenICam::gcstring](#) &PortName) const =0  
*Connects a port to a port node with given name.*
- virtual bool [Connect](#) ([IPort](#) \*pPort) const =0  
*Connects a port to the standard port "Device".*
- virtual void [Poll](#) (int64\_t ElapsedTime)=0  
*Fires nodes which have a polling time.*
- virtual [CLock](#) & [GetLock](#) () const =0  
*Returns the lock which guards the node map.*
- virtual uint64\_t [GetNumNodes](#) () const =0  
*Get the number of nodes in the map.*
- virtual void [LoadXMLFromFile](#) (const [GenICam::gcstring](#) &FileName)=0  
*Loads an XML from a file.*
- virtual void [LoadXMLFromFileInject](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectFileName)=0  
*Loads an XML from a file with injection.*
- virtual void [LoadXMLFromString](#) (const [GenICam::gcstring](#) &XMLData)=0  
*Loads an XML from a string.*
- virtual void [LoadXMLFromStringInject](#) (const [GenICam::gcstring](#) &TargetXMLData, const [GenICam::gcstring](#) &InjectXMLData)=0  
*Loads an XML from a string with injection.*
- virtual void [PreprocessXMLFromFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32\_t XMLValidation=[xv↵Default](#))=0  
*Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.*
- virtual void [MergeXMLFiles](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectedFileName, const [GenICam::gcstring](#) &OutputFileName)=0  
*Injects an XML file into a target file.*
- virtual void [ExtractIndependentSubtree](#) (const [GenICam::gcstring](#) &XMLData, const [GenICam::gcstring](#) &InjectXMLData, const [GenICam::gcstring](#) &SubTreeRootNodeName, [GenICam::gcstring](#) &Extracted↵Subtree)=0  
*Extract independent subtree.*
- virtual void [GetSupportedSchemaVersions](#) ([GenICam::gcstring\\_vector](#) &SchemaVersions)=0  
*Gets a list of supported schema versions.*
- virtual void [LoadXMLFromZIPFile](#) (const [GenICam::gcstring](#) &ZipFileName)=0  
*Loads an XML from a ZIP file.*
- virtual void [LoadXMLFromZIPData](#) (const void \*zipData, size\_t zipSize)=0  
*Loads an XML from a ZIP data buffer.*
- virtual void [PreprocessXMLFromZIPFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32\_t XMLValidation=[xv↵Default](#))=0  
*Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.*
- virtual void [Write](#) (const void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))=0  
*Writes a chunk of bytes to the port.*
- virtual [EYesNo](#) [GetSwapEndianness](#) ()=0  
*Determines if the port adapter must perform an endianness swap.*
- virtual void [Replay](#) ([IPort](#) \*pPort)=0

- Replays the write command to the given port interface.*

  - virtual void [SetCookie](#) (const int64\_t Value)=0

*Sets a cookie in case the port implementation want to cache a command list.*
- virtual int64\_t [GetCookie](#) ()=0

*Gets the cookie a port implementation may have set for caching a command list.*
- virtual void [StopRecording](#) ()=0

*Stops recording.*
- virtual void [Get](#) (uint8\_t \*pBuffer, int64\_t Length, bool Verify=false, bool IgnoreCache=false)=0

*Fills a buffer with the register's contents.*
- virtual int64\_t [GetLength](#) ()=0

*Retrieves the Length of the register [Bytes].*
- virtual int64\_t [GetAddress](#) ()=0

*Retrieves the Address of the register.*
- virtual void [GetSelectedFeatures](#) (FeatureList\_t &) const =0

*retrieve the group of selected features*
- virtual void [GetSelectingFeatures](#) (FeatureList\_t &) const =0

*retrieve the group of features selecting this node*
- virtual bool [SetNext](#) (bool Tick=true)=0

*Sets digit to next value.*
- virtual void [Restore](#) ()=0

*Restores the selectors' values found at creation.*
- virtual [GenICam::gcstring ToString](#) ()=0

*Returns a string representation of the digit.*
- virtual void [GetSelectorList](#) (FeatureList\_t &SelectorList, bool Incremental=false)=0

*Retrieves an ordered list of selectors.*
- virtual int64\_t [GetMaxLength](#) ()=0

*Retrieves the maximum length of the string in bytes.*
- virtual [GenICam::gcstring ToString](#) (bool Verify=false, bool IgnoreCache=false)=0

*Get content of the node as string.*
- virtual void [FromString](#) (const [GenICam::gcstring](#) &ValueStr, bool Verify=true)=0

*Set content of the node as string.*
- virtual bool [IsValueCacheValid](#) () const =0

*Checks if the value comes from cache or is requested from another node.*
- template<class Function >  
[CNodeCallback](#) \* [make\\_NodeCallback](#) (INode \*pNode, Function function, [ECallbackType](#) CallbackType)  
*make a new callback object for C functions*
- template<class Function >  
intptr\_t [Register](#) (INode \*pNode, Function f, [ECallbackType](#) CallbackType=[cbPostInsideLock](#))  
*Register a C-function as a callback.*
- template<class Client , class Member >  
[CNodeCallback](#) \* [make\\_NodeCallback](#) (INode \*pNode, Client &client, Member member, [ECallbackType](#) CallbackType)  
*make a new callback object for member functions*
- template<class Client , class Member >  
intptr\_t [Register](#) (INode \*pNode, Client &c, Member m, [ECallbackType](#) CallbackType=[cbPostInsideLock](#))  
*Register a C++-member function a callback.*
- [SPINNAKER\\_API](#) void [Deregister](#) ([GenApi::CallbackHandleType](#) pCallbackInfo)  
*Unregistering callback by handle.*
- [SPINNAKER\\_API](#) IDestroy \* [CastToIDestroy](#) (INodeMap \*pNodeMap)  
*makes sure the dynamic\_cast operator is implemented in the DLL (due to a Linux bug)*

- `template<class TCameraParams >`  
`void _LoadXMLFromFile (const GenICam::gcstring &FileName)`
- `template<class TCameraParams >`  
`void _LoadXMLFromZIPFile (const GenICam::gcstring &ZipFileName)`
- `template<class TCameraParams >`  
`void _LoadXMLFromFileInject (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)`
- `template<class TCameraParams >`  
`void _LoadXMLFromString (const GenICam::gcstring &XMLData)`
- `template<class TCameraParams >`  
`void _LoadXMLFromZIPData (const void *zipData, size_t zipSize)`
- `template<class TCameraParams >`  
`void _LoadXMLFromStringInject (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)`
- `template<class TCameraParams >`  
`void _GetSupportedSchemaVersions (GenICam::gcstring_vector &SchemaVersions)`
- `template<class TCameraParams >`  
`GenICam::gcstring _GetDeviceName ()`
- `template<class TCameraParams >`  
`void _Poll (int64_t ElapsedTime)`
- `template<class TCameraParams >`  
`void _GetNodes (NodeList_t &Nodes)`
- `template<class TCameraParams >`  
`INode * _GetNode (const GenICam::gcstring &key)`
- `template<class TCameraParams >`  
`void _InvalidateNodes ()`
- `template<class TCameraParams >`  
`bool _Connect (IPort *pPort, const GenICam::gcstring &PortName)`
- `template<class TCameraParams >`  
`bool _Connect (IPort *pPort)`
- `template<class TCameraParams >`  
`bool _ClearXMLCache ()`
- `virtual void PersistFeature (IValue &item)=0`  
*Stores a feature.*
- `SPINNAKER_API std::istream & EatComments (std::istream &is)`  
*Helper function ignoring lines starting with comment character '#'.  
Reads in persistent data from a stream.*
- `SPINNAKER_API std::istream & operator>> (std::istream &is, CFeatureBag &FeatureBag)`  
*Reads in persistent data from a stream.*
- `SPINNAKER_API std::ostream & operator<< (std::ostream &os, const CFeatureBag &FeatureBag)`  
*writes out persistent data to a stream*
- `template<class T, class B >`  
`bool IsReadable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`  
*Checks if a node is readable.*
- `template<class T, class B >`  
`bool IsWritable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`  
*Checks if a node is Writable.*
- `template<class T, class B >`  
`bool IsImplemented (const Spinnaker::GenApi::CPointer< T, B > &ptr)`  
*Checks if a node is Implemented.*
- `template<class T, class B >`  
`bool IsAvailable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`  
*Checks if a node is Available.*
- `GenICam::gcstring GetInterfaceName (IBase *pBase)`  
*Returns the name of the main interface as string DEPRICATED, use IBase::GetPrincipalInterfaceType() instead.*
- `virtual void SetNumEnums (int NumEnums)=0`  
*sets the number of enum values*

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IBase](#)  
*Base interface common to all nodes.*
- [const uint8\\_t COMMAND\\_MAGIC](#) = 0x42
- [const uint32\\_t U3V\\_EVENT\\_PREFIX](#) = 0x45563355
- [const uint16\\_t GENCP\\_EVENT\\_CMD\\_ID](#) = 0x0C00
- [const size\\_t GENCP\\_COMMAND\\_HEADER\\_SIZE](#) = sizeof(U3V\_COMMAND\_HEADER)
- [const size\\_t GENCP\\_EVENT\\_BASIC\\_SIZE](#) = sizeof(U3V\_EVENT\_MESSAGE)
- [interface SPINNAKER\\_API\\_ABSTRACT IBoolean](#)  
*Interface for Boolean properties.*
- [interface SPINNAKER\\_API\\_ABSTRACT bool Verify](#) = true) = 0
- [interface SPINNAKER\\_API\\_ABSTRACT ICategory](#)  
*Gives access to a category node.*
- [interface SPINNAKER\\_API\\_ABSTRACT IChunkPort](#)  
*Interface for ports attached to a chunk.*
- [interface SPINNAKER\\_API\\_ABSTRACT ICommand](#)  
*Interface for command like properties.*
- [interface SPINNAKER\\_API\\_ABSTRACT IDestroy](#)  
*Interface to destroy an object.*
- [interface SPINNAKER\\_API\\_ABSTRACT IDeviceInfo](#)  
*Interface to get information about the device (= nodemap)*
- [interface SPINNAKER\\_API\\_ABSTRACT IEnumEntry](#)  
*Interface of single enum value.*
- [interface SPINNAKER\\_API\\_ABSTRACT IEnumeration](#)  
*Interface for enumeration properties.*
- [template<typename EnumT > interface SPINNAKER\\_API\\_ABSTRACT IEnumerationT](#)  
*Interface for enumeration properties.*
- [template<typename EnumT > interface SPINNAKER\\_API\\_ABSTRACT virtual public IEnumReference](#)  
*Interface to construct an enum reference.*
- [interface SPINNAKER\\_API\\_ABSTRACT IFloat](#)  
*Interface for float properties.*
- [interface SPINNAKER\\_API\\_ABSTRACT IInteger](#)  
*Interface for integer properties.*
- [interface SPINNAKER\\_API\\_ABSTRACT INode](#)  
*Interface common to all nodes.*
- [interface SPINNAKER\\_API\\_ABSTRACT virtual public IReference](#)  
*Interface to construct a reference.*
- [interface SPINNAKER\\_API\\_ABSTRACT INodeMap](#)  
*Interface to access the node map.*
- [interface SPINNAKER\\_API\\_ABSTRACT INodeMapDyn](#)  
*Interface to access the node map.*
- [interface SPINNAKER\\_API\\_ABSTRACT IPort](#)  
*Interface for ports.*
- [interface SPINNAKER\\_API\\_ABSTRACT int64\\_t Address](#)
- [interface SPINNAKER\\_API\\_ABSTRACT int64\\_t int64\\_t Length](#) = 0
- [interface SPINNAKER\\_API IPortConstruct](#)  
*Interface for ports.*
- [interface SPINNAKER\\_API\\_ABSTRACT IPortWriteList](#)
- [interface SPINNAKER\\_API\\_ABSTRACT IPortReplay](#)

*Interface for replaying write commands on a port.*

- `interface SPINNAKER_API_ABSTRACT bool Invalidate = true) = 0`
- `interface SPINNAKER_API_ABSTRACT IPortRecorder`

*Interface for recording write commands on a port.*

- `interface SPINNAKER_API_ABSTRACT IRegister`

*Interface for registers.*

- `interface SPINNAKER_API_ABSTRACT ISelector`

*Interface for groups of features selected by a single one.*

- `interface SPINNAKER_API_ABSTRACT ISelectorDigit`

*Interface of a "digit" of the "counter" formed by the selector set.*

- `interface SPINNAKER_API_ABSTRACT IString`

*Interface for string properties.*

- `interface SPINNAKER_API_ABSTRACT IValue`

*Interface for value properties.*

- `interface SPINNAKER_API_ABSTRACT IPersistScript`

*Basic interface to persist values to.*

## 9.2.1 Typedef Documentation

9.2.1.1 `typedef IDevFileStreamBase<char, std::char_traits<char> > IDevFileStream`

9.2.1.2 `typedef ODevFileStreamBase<char, std::char_traits<char> > ODevFileStream`

## 9.2.2 Enumeration Type Documentation

9.2.2.1 `enum GVCP_MESSAGE_TAGS`

Enumerator

**`TAG_EVENT_CMD`**

**`TAG_EVENTDATA_CMD`**

## 9.2.3 Function Documentation

9.2.3.1 `virtual void Spinnaker::GenApi::PersistFeature ( IValue & item ) [pure virtual]`

Stores a feature.

9.2.3.2 void **SPINNAKER\_API** Spinnaker::GenApi::SET\_GUID ( SPIN\_GUID & *name*, uint32\_t *l*, uint16\_t *w1*, uint16\_t *w2*, uint8\_t *b1*, uint8\_t *b2*, uint8\_t *b3*, uint8\_t *b4*, uint8\_t *b5*, uint8\_t *b6*, uint8\_t *b7*, uint8\_t *b8* )

## 9.2.4 Variable Documentation

9.2.4.1 const uint8\_t **COMMAND\_MAGIC** = 0x42

9.2.4.2 const size\_t **GENCP\_COMMAND\_HEADER\_SIZE** = sizeof(U3V\_COMMAND\_HEADER)

9.2.4.3 const size\_t **GENCP\_EVENT\_BASIC\_SIZE** = sizeof(U3V\_EVENT\_MESSAGE)

9.2.4.4 const uint16\_t **GENCP\_EVENT\_CMD\_ID** = 0x0C00

9.2.4.5 interface **SPINNAKER\_API\_ABSTRACT** IPersistScript

**Initial value:**

```
{
    virtual void SetInfo(GenICam::gcstring &Info) = 0
}
```

Basic interface to persist values to.

9.2.4.6 const uint32\_t **U3V\_EVENT\_PREFIX** = 0x45563355

## 9.3 Spinnaker::GenICam Namespace Reference

### Classes

- class [AutoLock](#)
- class [CGlobalLock](#)
  - Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)
  - Unlocks the global lock object on destruction.*
- class [CLock](#)
  - A lock class.*
- class [CLockEx](#)
  - This class is for testing purposes only.*
- class [gcstring](#)
- class [LockableObject](#)
  - Instance-Lock for an object.*
- struct [Version\\_t](#)
  - Version.*



## Functions

- [SPINNAKER\\_API](#) void [ThrowBadAlloc](#) ()
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenICam::gcstring](#) &str)  
*STL getline.*
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenICam::gcstring](#) &str, char delim)  
*STL getline.*
- template<typename Td , typename Ts >  
Td [INTEGRAL\\_CAST2](#) (Ts s)  
*This verifies at runtime if there was no loss of data if an type Ts (e.g.*
- template<typename T >  
T [INTEGRAL\\_CAST](#) (int64\_t ll)  
*This verifies at runtime if there was no loss of data if an int64\_t was downcast to type T (e.g.*
- [SPINNAKER\\_API](#) bool [DoesEnvironmentVariableExist](#) (const [Spinnaker::GenICam::gcstring](#) &VariableName)  
*Returns true if an environment variable exists.*
- [SPINNAKER\\_API](#) [gcstring](#) [GetValueOfEnvironmentVariable](#) (const [gcstring](#) &VariableName)  
*Retrieve the value of an environment variable.*
- [SPINNAKER\\_API](#) bool [GetValueOfEnvironmentVariable](#) (const [gcstring](#) &VariableName, [gcstring](#) &VariableContent)  
*Retrieve the value of an environment variable.*
- [SPINNAKER\\_API](#) [gcstring](#) [UrlEncode](#) (const [gcstring](#) &Input)  
*Converts \ to / and replaces all unsafe characters by their xx equivalent.*
- [SPINNAKER\\_API](#) [gcstring](#) [UrlDecode](#) (const [gcstring](#) &Input)  
*Replaces xx escapes by their char equivalent.*
- [SPINNAKER\\_API](#) void [ReplaceEnvironmentVariables](#) ([gcstring](#) &Buffer, bool ReplaceBlankBy20=false)  
*Replaces in a string and replace ' ' with %20.*
- [SPINNAKER\\_API](#) [gcstring](#) [GetGenICamCacheFolder](#) (void)  
*Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling [SetGenICamCacheFolder\(\)](#).*
- [SPINNAKER\\_API](#) [gcstring](#) [GetGenICamLogConfig](#) (void)  
*Retrieve the path of the GenICam logging properties file.*
- [SPINNAKER\\_API](#) [gcstring](#) [GetGenICamCLProtocolFolder](#) (void)  
*Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling [SetGenICamCLProtocolFolder\(\)](#).*
- [SPINNAKER\\_API](#) void [SetGenICamCacheFolder](#) (const [gcstring](#) &path)  
*Stores the path of the GenICam cache folder.*
- [SPINNAKER\\_API](#) void [SetGenICamLogConfig](#) (const [gcstring](#) &path)  
*Stores the path of the GenICam logging properties file.*
- [SPINNAKER\\_API](#) void [SetGenICamCLProtocolFolder](#) (const [gcstring](#) &path)  
*Stores the path of the CLProtocol folder.*
- [SPINNAKER\\_API](#) void [Tokenize](#) (const [gcstring](#) &str, [gcstring\\_vector](#) &tokens, const [gcstring](#) &delimiters="")  
*splits str input string into a list of tokens using the delimiter*
- [SPINNAKER\\_API](#) void [GetFiles](#) (const [gcstring](#) &FileTemplate, [gcstring\\_vector](#) &FileNames, const bool DirectoriesOnly=false)  
*Gets a list of files or directories matching a given FileTemplate.*
- [SPINNAKER\\_API](#) [gcstring](#) [GetModulePathFromFunction](#) (void \*pFunction)  
*Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.*

### 9.3.1 Function Documentation

9.3.1.1 `std::istream& Spinnaker::GenICam::getline ( std::istream & is, Spinnaker::GenICam::gcstring & str )`  
[inline]

STL getline.

9.3.1.2 `std::istream& Spinnaker::GenICam::getline ( std::istream & is, Spinnaker::GenICam::gcstring & str, char delim )`  
[inline]

STL getline.

9.3.1.3 `SPINNAKER_API void Spinnaker::GenICam::ThrowBadAlloc ( )`

## 9.4 Spinnaker::Video Namespace Reference

### Classes

- struct [AVIOption](#)  
*Options for saving AVI files.*
- struct [H264Option](#)  
*Options for saving H264 files.*
- struct [MJPGOption](#)  
*Options for saving MJPG files.*
- class [SpinVideo](#)  
*Provides the functionality for the user to record images to an AVI/MP4 file.*

## Chapter 10

# Class Documentation

### 10.1 ActionCommandResult Struct Reference

Action Command Result.

#### Public Attributes

- unsigned int [DeviceAddress](#)
- [ActionCommandStatus](#) Status

#### 10.1.1 Detailed Description

Action Command Result.

#### 10.1.2 Member Data Documentation

10.1.2.1 unsigned int [DeviceAddress](#)

10.1.2.2 [ActionCommandStatus](#) Status

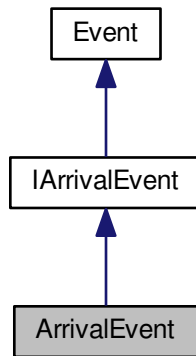
The documentation for this struct was generated from the following file:

- include/[SpinnakerDefs.h](#)

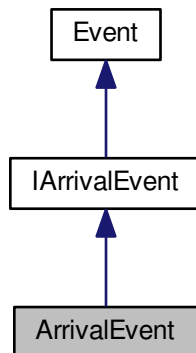
## 10.2 ArrivalEvent Class Reference

An event handler for capturing the device arrival event.

Inheritance diagram for ArrivalEvent:



Collaboration diagram for ArrivalEvent:



### Public Member Functions

- [ArrivalEvent](#) ()  
*Default constructor.*
- virtual [~ArrivalEvent](#) ()  
*Virtual destructor.*
- virtual void [OnDeviceArrival](#) (uint64\_t serialNumber)=0  
*Callback to the device arrival event.*

## Protected Member Functions

- [ArrivalEvent](#) & `operator=` (const [ArrivalEvent](#) &)  
*Assignment operator.*

## Additional Inherited Members

### 10.2.1 Detailed Description

An event handler for capturing the device arrival event.

### 10.2.2 Constructor & Destructor Documentation

#### 10.2.2.1 `ArrivalEvent ( )`

Default constructor.

#### 10.2.2.2 `virtual ~ArrivalEvent ( ) [virtual]`

Virtual destructor.

### 10.2.3 Member Function Documentation

#### 10.2.3.1 `virtual void OnDeviceArrival ( uint64_t serialNumber ) [pure virtual]`

Callback to the device arrival event.

Implements [IArrivalEvent](#).

#### 10.2.3.2 `ArrivalEvent& operator= ( const ArrivalEvent & ) [protected]`

Assignment operator.

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

- `include/ArrivalEvent.h`

## 10.3 AttachStatistics\_t Struct Reference

Delivers information about the attached chunks and nodes.

## Public Attributes

- int [NumChunkPorts](#)  
*Number of chunk ports found in the node map.*
- int [NumChunks](#)  
*Number of chunks found in the buffer.*
- int [NumAttachedChunks](#)  
*Number of chunks from the buffer attached to a chunk port.*

### 10.3.1 Detailed Description

Delivers information about the attached chunks and nodes.

### 10.3.2 Member Data Documentation

#### 10.3.2.1 int NumAttachedChunks

Number of chunks from the buffer attached to a chunk port.

#### 10.3.2.2 int NumChunkPorts

Number of chunk ports found in the node map.

#### 10.3.2.3 int NumChunks

Number of chunks found in the buffer.

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

- include/SpinGenApi/[ChunkAdapter.h](#)

## 10.4 AutoLock Class Reference

### Public Member Functions

- [AutoLock](#) ([CLOCK](#) &lock)
- [~AutoLock](#) ()

### 10.4.1 Constructor & Destructor Documentation

#### 10.4.1.1 [AutoLock](#) ( [CLOCK](#) & lock ) [inline]

#### 10.4.1.2 [~AutoLock](#) ( ) [inline]

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

- include/SpinGenApi/[Synch.h](#)

## 10.5 AutoLock Class Reference

### Public Member Functions

- [AutoLock](#) ([CLOCK](#) &lock)
- [~AutoLock](#) ()

### 10.5.1 Constructor & Destructor Documentation

10.5.1.1 [AutoLock](#) ( [CLOCK](#) & lock ) [inline]

10.5.1.2 [~AutoLock](#) ( ) [inline]

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

- include/SpinGenApi/[GCSynch.h](#)

## 10.6 AVIOption Struct Reference

Options for saving AVI files.

### Public Member Functions

- [AVIOption](#) ()

### Public Attributes

- float [frameRate](#)  
*Frame rate of the stream.*
- unsigned int [reserved](#) [256]  
*Reserved for future use.*

### 10.6.1 Detailed Description

Options for saving AVI files.

### 10.6.2 Constructor & Destructor Documentation

10.6.2.1 [AVIOption](#) ( ) [inline]

### 10.6.3 Member Data Documentation

10.6.3.1 float [frameRate](#)

Frame rate of the stream.

### 10.6.3.2 unsigned int reserved[256]

Reserved for future use.

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

- include/[SpinVideoDefs.h](#)

## 10.7 BasePtr< T, B > Class Template Reference

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

### Public Member Functions

- [BasePtr](#) (void) throw ()  
*Default constructor.*
- virtual [~BasePtr](#) (void)
- [BasePtr](#) (const [BasePtr](#) &other) throw ()
- virtual [BasePtr](#) & [operator=](#) (const [BasePtr](#) &rhs)  
*Assign INode Pointer.*
- virtual [BasePtr](#) & [operator=](#) (const int nMustBeNull)
- virtual [operator T \\*](#) (void) const  
*Dereferencing.*
- virtual T \* [operator->](#) (void) const  
*Dereferencing.*
- virtual bool [IsValid](#) () const throw ()  
*True if the pointer is valid.*
- virtual [operator bool](#) (void) const throw ()  
*True if the pointer is valid.*
- virtual bool [operator==](#) (const [BasePtr](#) &rT) const  
*Pointer equal.*
- virtual bool [operator==](#) (int nMustBeNull) const  
*Pointer equal.*
- virtual T \* [get](#) () const  
[get\(\)](#)

### Protected Attributes

- PointerData \* [m\\_pT](#)  
*Underlying raw pointer.*

### 10.7.1 Detailed Description

```
template<class T, class B = T>
class Spinnaker::BasePtr< T, B >
```

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.



## 10.7.2 Constructor & Destructor Documentation

### 10.7.2.1 BasePtr ( void ) throw )

Default constructor.

### 10.7.2.2 virtual ~BasePtr ( void ) [virtual]

### 10.7.2.3 BasePtr ( const BasePtr< T, B > & other ) throw )

## 10.7.3 Member Function Documentation

### 10.7.3.1 virtual T\* get ( ) const [virtual]

[get\(\)](#)

### 10.7.3.2 virtual bool isValid ( ) const throw ) [virtual]

True if the pointer is valid.

### 10.7.3.3 virtual operator bool ( void ) const throw ) [virtual]

True if the pointer is valid.

### 10.7.3.4 virtual operator T\* ( void ) const [virtual]

Dereferencing.

### 10.7.3.5 virtual T\* operator-> ( void ) const [virtual]

Dereferencing.

### 10.7.3.6 virtual BasePtr& operator= ( const BasePtr< T, B > & rhs ) [virtual]

Assign INode Pointer.

### 10.7.3.7 virtual BasePtr& operator= ( const int nMustBeNull ) [virtual]

Reimplemented in [ImagePtr](#), [InterfacePtr](#), [LoggingEventDataPtr](#), and [CameraPtr](#).

10.7.3.8 `virtual bool operator==( const BasePtr< T, B > & rT ) const` `[virtual]`

Pointer equal.

10.7.3.9 `virtual bool operator==( int nMustBeNull ) const` `[virtual]`

Pointer equal.

## 10.7.4 Member Data Documentation

10.7.4.1 `PointerData* m_pT` `[protected]`

Underlying raw pointer.

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

- include/[BasePtr.h](#)

## 10.8 BMPOption Struct Reference

Options for saving Bitmap image.

### Public Member Functions

- [BMPOption](#) ()

### Public Attributes

- bool [indexedColor\\_8bit](#)
- unsigned int [reserved](#) [16]

*Reserved for future use.*

### 10.8.1 Detailed Description

Options for saving Bitmap image.

## 10.8.2 Constructor & Destructor Documentation

10.8.2.1 `BMPOption( )` `[inline]`

## 10.8.3 Member Data Documentation

10.8.3.1 `bool indexedColor_8bit`

10.8.3.2 `unsigned int reserved[16]`

Reserved for future use.

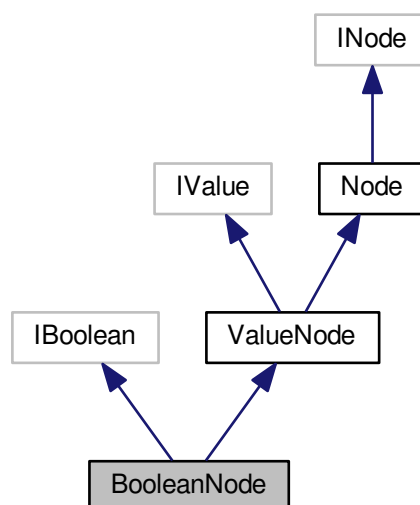
The documentation for this struct was generated from the following file:

- `include/SpinnakerDefs.h`

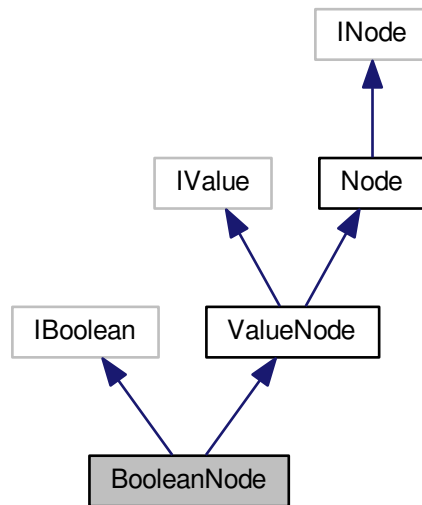
## 10.9 BooleanNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for BooleanNode:



Collaboration diagram for BooleanNode:



## Public Member Functions

- [BooleanNode](#) ()
- [BooleanNode](#) (std::shared\_ptr< Node::NodeImpl > pBoolean)
- virtual [~BooleanNode](#) ()
- void [SetValue](#) (bool Value, bool [Verify](#)=true)  
*Set node value.*
- virtual void [operator=](#) (bool Value)  
*Set node value.*
- bool [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false) const  
*Get node value.*
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.9.1 Detailed Description

[Interface](#) for string properties.

## 10.9.2 Constructor & Destructor Documentation

### 10.9.2.1 BooleanNode ( )

### 10.9.2.2 BooleanNode ( std::shared\_ptr< Node::NodeImpl > *pBoolean* )

### 10.9.2.3 virtual ~BooleanNode ( ) [virtual]

## 10.9.3 Member Function Documentation

### 10.9.3.1 bool GetValue ( bool *Verify* = false, bool *IgnoreCache* = false ) const

Get node value.

#### Parameters

|                    |                                                                                 |
|--------------------|---------------------------------------------------------------------------------|
| <i>Verify</i>      | Enables Range verification (default = false). The AccessMode is always checked. |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false).                |

#### Returns

The value read.

### 10.9.3.2 virtual void operator= ( bool *Value* ) [virtual]

Set node value.

### 10.9.3.3 virtual void SetReference ( INode \* *pBase* ) [virtual]

overload SetReference for Value

Reimplemented from [ValueNode](#).

### 10.9.3.4 void SetValue ( bool *Value*, bool *Verify* = true )

Set node value.

#### Parameters

|               |                                                             |
|---------------|-------------------------------------------------------------|
| <i>Value</i>  | The value to set.                                           |
| <i>Verify</i> | Enables AccessMode and Range verification (default = true). |

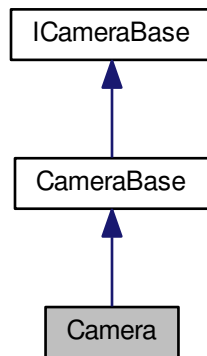
The documentation for this class was generated from the following file:

- include/SpinGenApi/[BooleanNode.h](#)

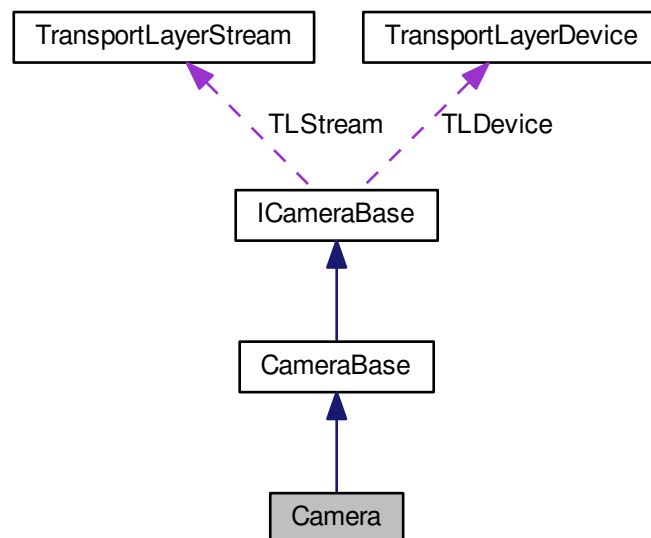
## 10.10 Camera Class Reference

The camera object class.

Inheritance diagram for Camera:



Collaboration diagram for Camera:



### Public Member Functions

- [~Camera](#) ()
- void [Init](#) ()

## Public Attributes

- [GenApi::Integer](#) & [LUTIndex](#)  
Description:  
*Control the index (offset) of the coefficient to access in the selected LUT.*
- [GenApi::Boolean](#) & [LUTEnable](#)  
Description:  
*Activates the selected LUT.*
- [GenApi::Integer](#) & [LUTValue](#)  
Description:  
*Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.*
- [GenApi::EnumerationT< LUTSelectorEnums >](#) & [LUTSelector](#)  
Description:  
*Selects which LUT to control.*
- [GenApi::Float](#) & [ExposureTime](#)  
Description:  
*Exposure time in microseconds when Exposure Mode is Timed.*
- [GenApi::Command](#) & [AcquisitionStop](#)  
Description: *This command stops the acquisition of images.*
- [GenApi::Float](#) & [AcquisitionResultingFrameRate](#)  
Description: *Resulting frame rate in Hertz.*
- [GenApi::Float](#) & [AcquisitionLineRate](#)  
Description: *Controls the rate (in Hertz) at which the Lines in a Frame are captured.*
- [GenApi::Command](#) & [AcquisitionStart](#)  
Description: *This command starts the acquisition of images.*
- [GenApi::Command](#) & [TriggerSoftware](#)  
Description:  
*Generates an internal trigger if Trigger Source is set to Software.*
- [GenApi::EnumerationT< ExposureModeEnums >](#) & [ExposureMode](#)  
Description:  
*Sets the operation mode of the Exposure.*
- [GenApi::EnumerationT< AcquisitionModeEnums >](#) & [AcquisitionMode](#)  
Description: *Sets the acquisition mode of the device.*
- [GenApi::Integer](#) & [AcquisitionFrameCount](#)  
Description:  
*Number of images to acquire during a multi frame acquisition.*
- [GenApi::EnumerationT< TriggerSourceEnums >](#) & [TriggerSource](#)  
Description:  
*Specifies the internal signal or physical input line to use as the trigger source.*
- [GenApi::EnumerationT< TriggerActivationEnums >](#) & [TriggerActivation](#)  
Description: *Specifies the activation mode of the trigger.*
- [GenApi::EnumerationT< SensorShutterModeEnums >](#) & [SensorShutterMode](#)  
Description: *Sets the shutter mode of the device.*
- [GenApi::Float](#) & [TriggerDelay](#)  
Description:  
*Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.*
- [GenApi::EnumerationT< TriggerModeEnums >](#) & [TriggerMode](#)  
Description:  
*Controls whether or not trigger is active.*
- [GenApi::Float](#) & [AcquisitionFrameRate](#)  
Description: *User controlled acquisition frame rate in Hertz Visibility:*
- [GenApi::EnumerationT< TriggerOverlapEnums >](#) & [TriggerOverlap](#)  
Description: *Specifies the overlap mode of the trigger.*
- [GenApi::EnumerationT< TriggerSelectorEnums >](#) & [TriggerSelector](#)

- Description: Selects the type of trigger to configure.*

  - [GenApi::IBoolean](#) & [AcquisitionFrameRateEnable](#)

*Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.*

- [GenApi::IEnumerationT< ExposureAutoEnums >](#) & [ExposureAuto](#)

*Description: Sets the automatic exposure mode Visibility:*

- [GenApi::Integer](#) & [AcquisitionBurstFrameCount](#)

*Description:*

*This feature is used only if the FrameBurstStart trigger is enabled and the FrameBurstEnd trigger is disabled.*

- [GenApi::Integer](#) & [EventTest](#)

*Description: Returns the unique identifier of the Test type of [Event](#).*

- [GenApi::Integer](#) & [EventTestTimestamp](#)

*Description: Returns the Timestamp of the Test [Event](#).*

- [GenApi::Integer](#) & [EventExposureEndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End [Event](#).*

- [GenApi::Integer](#) & [EventExposureEnd](#)

*Description: Returns the unique identifier of the Exposure End type of [Event](#).*

- [GenApi::Integer](#) & [EventExposureEndTimestamp](#)

*Description: Returns the Timestamp of the Exposure End [Event](#).*

- [GenApi::Integer](#) & [EventError](#)

*Description: Returns the unique identifier of the Error type of [Event](#).*

- [GenApi::Integer](#) & [EventErrorTimestamp](#)

*Description: Returns the Timestamp of the Error [Event](#).*

- [GenApi::Integer](#) & [EventErrorCode](#)

*Description: Returns the error code for the error that happened Visibility:*

- [GenApi::Integer](#) & [EventErrorFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Error [Event](#).*

- [GenApi::IEnumerationT< EventSelectorEnums >](#) & [EventSelector](#)

*Description: Selects which [Event](#) to enable or disable.*

- [GenApi::IBoolean](#) & [EventSerialReceiveOverflow](#)

*Description: Returns the status of the event serial receive overflow.*

- [GenApi::Integer](#) & [EventSerialPortReceive](#)

*Description: Returns the unique identifier of the Serial Port Receive type of [Event](#).*

- [GenApi::Integer](#) & [EventSerialPortReceiveTimestamp](#)

*Description: Returns the Timestamp of the Serial Port Receive [Event](#).*

- [GenApi::IString](#) & [EventSerialData](#)

*Description: Returns the serial data that was received.*

- [GenApi::Integer](#) & [EventSerialDataLength](#)

*Description: Returns the length of the received serial data that was included in the event payload.*

- [GenApi::IEnumerationT< EventNotificationEnums >](#) & [EventNotification](#)

*Description: Enables/Disables the selected event.*

- [GenApi::Integer](#) & [LogicBlockLUTRowIndex](#)

*Description: Controls the row of the truth table to access in the selected LUT.*

- [GenApi::IEnumerationT< LogicBlockSelectorEnums >](#) & [LogicBlockSelector](#)

*Description: Selects which LogicBlock to configure Visibility:*

- [GenApi::IEnumerationT< LogicBlockLUTInputActivationEnums >](#) & [LogicBlockLUTInputActivation](#)

*Description: Selects the activation mode of the Logic Input Source signal.*

- [GenApi::IEnumerationT< LogicBlockLUTInputSelectorEnums >](#) & [LogicBlockLUTInputSelector](#)

*Description: Controls which LogicBlockLUT Input Source & Activation to access.*

- [GenApi::IEnumerationT< LogicBlockLUTInputSourceEnums >](#) & [LogicBlockLUTInputSource](#)

*Description: Selects the source for the input into the Logic LUT.*

- [GenApi::IBoolean](#) & [LogicBlockLUTOutputValue](#)



- Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.*

  - [GenApi::Integer](#) & [LogicBlockLUTOutputValueAll](#)

*Description: Sets the value of all the output bits in the selected LUT.*

- [GenApi::EnumerationT](#) < [LogicBlockLUTSelectorEnums](#) > & [LogicBlockLUTSelector](#)

*Description: Selects which LogicBlock LUT to configure Visibility:*

- [GenApi::IFloat](#) & [ColorTransformationValue](#)

*Description:*  
*Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.*

- [GenApi::IBoolean](#) & [ColorTransformationEnable](#)

*Description:*  
*Enables/disables the color transform selected with ColorTransformationSelector.*

- [GenApi::EnumerationT](#) < [ColorTransformationSelectorEnums](#) > & [ColorTransformationSelector](#)

*Description: Selects which Color Transformation module is controlled by the various Color Transformation features.*

- [GenApi::EnumerationT](#) < [RgbTransformLightSourceEnums](#) > & [RgbTransformLightSource](#)

*Description:*  
*Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources.*

- [GenApi::IFloat](#) & [Saturation](#)

*Description: Controls the color saturation.*

- [GenApi::IBoolean](#) & [SaturationEnable](#)

*Description: Enables/disables Saturation adjustment.*

- [GenApi::EnumerationT](#) < [ColorTransformationValueSelectorEnums](#) > & [ColorTransformationValueSelector](#)

*Description:*  
*Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module*

- [GenApi::Integer](#) & [TimestampLatchValue](#)

*Description: Returns the latched value of the timestamp counter.*

- [GenApi::Command](#) & [TimestampReset](#)

*Description: Resets the current value of the device timestamp counter.*

- [GenApi::IString](#) & [DeviceUserID](#)

*Description: User-programmable device identifier.*

- [GenApi::IFloat](#) & [DeviceTemperature](#)

*Description: Device temperature in degrees Celsius (C).*

- [GenApi::Integer](#) & [MaxDeviceResetTime](#)

*Description: Time to wait until device reset complete (ms).*

- [GenApi::Integer](#) & [DeviceTLVersionMinor](#)

*Description:*  
*Minor version of the Transport Layer of the device.*

- [GenApi::IString](#) & [DeviceSerialNumber](#)

*Description:*  
*Device's serial number.*

- [GenApi::IString](#) & [DeviceVendorName](#)

*Description: Name of the manufacturer of the device.*

- [GenApi::EnumerationT](#) < [DeviceRegistersEndiannessEnums](#) > & [DeviceRegistersEndianness](#)

*Description: Endianness of the registers of the device.*

- [GenApi::IString](#) & [DeviceManufacturerInfo](#)

*Description: Manufacturer information about the device.*

- [GenApi::Integer](#) & [DeviceLinkSpeed](#)

*Description:*  
*Indicates the speed of transmission negotiated on the specified Link.*

- [GenApi::Integer](#) & [LinkUptime](#)

*Description: Time since the last phy negotiation (enumeration).*

- [GenApi::Integer](#) & [DeviceEventChannelCount](#)

*Description:*  
*Indicates the number of event channels supported by the device.*

- [GenApi::ICommand](#) & [TimestampLatch](#)  
*Description: Latches the current timestamp counter into TimestampLatchValue.*
- [GenApi::IEnumerationT< DeviceScanTypeEnums >](#) & [DeviceScanType](#)  
*Description: Scan type of the sensor of the device.*
- [GenApi::ICommand](#) & [DeviceReset](#)  
*Description: This is a command that immediately resets and reboots the device.*
- [GenApi::IEnumerationT< DeviceCharacterSetEnums >](#) & [DeviceCharacterSet](#)  
*Description:*  
*Character set used by the strings of the device's bootstrap registers.*
- [GenApi::Integer](#) & [DeviceLinkThroughputLimit](#)  
*Description:*  
*Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link.*
- [GenApi::IString](#) & [DeviceFirmwareVersion](#)  
*Description: Version of the firmware on the device.*
- [GenApi::Integer](#) & [DeviceStreamChannelCount](#)  
*Description:*  
*Indicates the number of streaming channels supported by the device.*
- [GenApi::IEnumerationT< DeviceTLTypeEnums >](#) & [DeviceTLType](#)  
*Description: Transport Layer type of the device.*
- [GenApi::IString](#) & [DeviceVersion](#)  
*Description: Version of the device.*
- [GenApi::IEnumerationT< DevicePowerSupplySelectorEnums >](#) & [DevicePowerSupplySelector](#)  
*Description:*  
*Selects the power supply source to control or read.*
- [GenApi::IString](#) & [SensorDescription](#)  
*Description: Returns Sensor Description Visibility:*
- [GenApi::IString](#) & [DeviceModelName](#)  
*Description: Model of the device.*
- [GenApi::Integer](#) & [DeviceTLVersionMajor](#)  
*Description:*  
*Major version of the Transport Layer of the device.*
- [GenApi::IEnumerationT< DeviceTemperatureSelectorEnums >](#) & [DeviceTemperatureSelector](#)  
*Description:*  
*Selects the location within the device, where the temperature will be measured.*
- [GenApi::Integer](#) & [EnumerationCount](#)  
*Description: Number of enumerations since uptime.*
- [GenApi::IFloat](#) & [PowerSupplyCurrent](#)  
*Description:*  
*Indicates the output current of the selected power supply (A).*
- [GenApi::IString](#) & [DeviceID](#)  
*Description: Device identifier (serial number).*
- [GenApi::Integer](#) & [DeviceUptime](#)  
*Description: Total time since the device was powered up in seconds.*
- [GenApi::Integer](#) & [DeviceLinkCurrentThroughput](#)  
*Description: Current bandwidth of streamed data.*
- [GenApi::Integer](#) & [DeviceMaxThroughput](#)  
*Description:*  
*Maximum bandwidth of the data that can be streamed out of the device.*
- [GenApi::ICommand](#) & [FactoryReset](#)  
*Description: Returns all user tables to factory default Visibility:*
- [GenApi::IFloat](#) & [PowerSupplyVoltage](#)  
*Description:*  
*Indicates the current voltage of the selected power supply (V).*

- [GenApi::IEnumerationT < DeviceIndicatorModeEnums > & DeviceIndicatorMode](#)  
*Description: Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).*
- [GenApi::IFloat & DeviceLinkBandwidthReserve](#)  
*Description: Percentage of streamed data bandwidth reserved for packet resend.*
- [GenApi::Integer & AasRoiOffsetY](#)  
*Description: Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::Integer & AasRoiOffsetX](#)  
*Description: Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::IEnumerationT < AutoExposureControlPriorityEnums > & AutoExposureControlPriority](#)  
*Description: Selects whether to adjust gain or exposure first.*
- [GenApi::IFloat & BalanceWhiteAutoLowerLimit](#)  
*Description: Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.*
- [GenApi::IFloat & BalanceWhiteAutoDamping](#)  
*Description: Controls how quickly 'BalanceWhiteAuto' adjusts the values for Red and Blue BalanceRatio in response to changing conditions.*
- [GenApi::Integer & AasRoiHeight](#)  
*Description: Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::IFloat & AutoExposureGreyValueUpperLimit](#)  
*Description: The highest value in percentage that the target mean may reach.*
- [GenApi::IFloat & AutoExposureTargetGreyValue](#)  
*Description: This is the user-specified target grey level (image mean) to apply to the current image.*
- [GenApi::IFloat & AutoExposureGainLowerLimit](#)  
*Description: The smallest gain that auto exposure can set.*
- [GenApi::IFloat & AutoExposureGreyValueLowerLimit](#)  
*Description: The lowest value in percentage that the target mean may reach.*
- [GenApi::IEnumerationT < AutoExposureMeteringModeEnums > & AutoExposureMeteringMode](#)  
*Description: Selects a metering mode: average, spot, or partial metering.*
- [GenApi::IFloat & AutoExposureExposureTimeUpperLimit](#)  
*Description: The largest exposure time that auto exposure can set.*
- [GenApi::IFloat & AutoExposureGainUpperLimit](#)  
*Description: The largest gain that auto exposure can set.*
- [GenApi::IFloat & AutoExposureControlLoopDamping](#)  
*Description: It controls how fast the exposure and gain get settled.*
- [GenApi::IFloat & AutoExposureEVCompensation](#)  
*Description: The EV compensation value used in the exposure compensation.*
- [GenApi::IFloat & AutoExposureExposureTimeLowerLimit](#)

Description:

*The smallest exposure time that auto exposure can set.*

- [GenApi::IEnumerationT](#) < [BalanceWhiteAutoProfileEnums](#) > & [BalanceWhiteAutoProfile](#)

Description: *Selects the profile used by BalanceWhiteAuto.*

- [GenApi::IEnumerationT](#) < [AutoAlgorithmSelectorEnums](#) > & [AutoAlgorithmSelector](#)

Description:

*Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.*

- [GenApi::IEnumerationT](#) < [AutoExposureTargetGreyValueAutoEnums](#) > & [AutoExposureTargetGreyValue↔Auto](#)

Description:

*This indicates whether the target image grey level is automatically set by the camera or manually set by the user.*

- [GenApi::IBoolean](#) & [AasRoiEnable](#)

Description:

*Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the AutoAlgorithm↔Selector feature.*

- [GenApi::IEnumerationT](#) < [AutoExposureLightingModeEnums](#) > & [AutoExposureLightingMode](#)

Description:

*Selects a lighting mode: Backlight, Frontlight or Normal (default).*

- [GenApi::IInteger](#) & [AasRoiWidth](#)

Description:

*Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*

- [GenApi::IFloat](#) & [BalanceWhiteAutoUpperLimit](#)

Description:

*Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.*

- [GenApi::IInteger](#) & [LinkErrorCount](#)

Description: *Counts the number of error on the link.*

- [GenApi::IBoolean](#) & [GevCurrentIPConfigurationDHCP](#)

Description: *Controls whether the DHCP IP configuration scheme is activated on the given logical link.*

- [GenApi::IInteger](#) & [GevInterfaceSelector](#)

Description: *Selects which logical link to control.*

- [GenApi::IInteger](#) & [GevSCPD](#)

Description: *Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.*

- [GenApi::IInteger](#) & [GevTimestampTickFrequency](#)

Description: *Indicates the number of timestamp ticks in 1 second (frequency in Hz).*

- [GenApi::IInteger](#) & [GevSCPSPacketSize](#)

Description: *Specifies the stream packet size (in bytes) to send on this channel.*

- [GenApi::IInteger](#) & [GevCurrentDefaultGateway](#)

Description: *Reports the default gateway IP address to be used on the given logical link.*

- [GenApi::IBoolean](#) & [GevSCCFGUnconditionalStreaming](#)

Description: *Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).*

- [GenApi::IInteger](#) & [GevMCTT](#)

Description: *Indicates the transmission timeout of the message channel.*

- [GenApi::IBoolean](#) & [GevSCPSPDoNotFragment](#)

Description: *The state of this feature is copied into the "do not fragment" bit of the IP header of each stream packet.*

- [GenApi::IInteger](#) & [GevCurrentSubnetMask](#)

Description: *Reports the subnet mask of the given logical link.*

- [GenApi::IInteger](#) & [GevStreamChannelSelector](#)

Description: *Selects the stream channel to control.*

- [GenApi::IInteger](#) & [GevCurrentIPAddress](#)

Description: *Reports the IP address for the given logical link.*

- [GenApi::IInteger](#) & [GevMCSP](#)

- Description: Indicates the source port of the message channel.*

  - [GenApi::Integer](#) & [GevGVCPPendingTimeout](#)

*Description: Indicates the longest GVCP command execution time before the device returns a PENDING\_ACK in milliseconds.*
- [GenApi::EnumerationT](#) < [GevIEEE1588StatusEnums](#) > & [GevIEEE1588Status](#)

*Description: Provides the status of the IEEE 1588 clock.*
- [GenApi::IString](#) & [GevFirstURL](#)

*Description: The first choice of URL for the XML device description file.*
- [GenApi::Integer](#) & [GevMACAddress](#)

*Description: MAC address of the logical link.*
- [GenApi::Integer](#) & [GevPersistentSubnetMask](#)

*Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.*
- [GenApi::Integer](#) & [GevMCPHostPort](#)

*Description: The port to which the device must send messages Visibility:*
- [GenApi::Integer](#) & [GevSCPHostPort](#)

*Description: Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.*
- [GenApi::IBoolean](#) & [GevGVCPPendingAck](#)

*Description: Enables the generation of PENDING\_ACK.*
- [GenApi::Integer](#) & [GevSCPIInterfaceIndex](#)

*Description: Index of the logical link to use.*
- [GenApi::IBoolean](#) & [GevSupportedOption](#)

*Description: Returns if the selected GEV option is supported.*
- [GenApi::EnumerationT](#) < [GevIEEE1588ModeEnums](#) > & [GevIEEE1588Mode](#)

*Description: Provides the mode of the IEEE 1588 clock.*
- [GenApi::IBoolean](#) & [GevCurrentIPConfigurationLLA](#)

*Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.*
- [GenApi::Integer](#) & [GevSCSP](#)

*Description: Indicates the source port of the stream channel.*
- [GenApi::IBoolean](#) & [GevIEEE1588](#)

*Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.*
- [GenApi::IBoolean](#) & [GevSCCFGExtendedChunkData](#)

*Description: Enables cameras to use the extended chunk data payload type for this stream channel.*
- [GenApi::Integer](#) & [GevPersistentIPAddress](#)

*Description: Controls the Persistent IP address for this logical link.*
- [GenApi::IBoolean](#) & [GevCurrentIPConfigurationPersistentIP](#)

*Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.*
- [GenApi::EnumerationT](#) < [GevIEEE1588ClockAccuracyEnums](#) > & [GevIEEE1588ClockAccuracy](#)

*Description: Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.*
- [GenApi::Integer](#) & [GevHeartbeatTimeout](#)

*Description: Indicates the current heartbeat timeout in milliseconds.*
- [GenApi::Integer](#) & [GevPersistentDefaultGateway](#)

*Description: Controls the persistent default gateway for this logical link.*
- [GenApi::EnumerationT](#) < [GevCCPEnums](#) > & [GevCCP](#)

*Description: Controls the device access privilege of an application.*
- [GenApi::Integer](#) & [GevMCDA](#)

*Description: Controls the destination IP address of the message channel Visibility:*
- [GenApi::Integer](#) & [GevSCDA](#)

*Description: Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.*

- [GenApi::Integer](#) & [GevSCPDirection](#)  
Description: Transmit or Receive of the channel Visibility:
- [GenApi::Boolean](#) & [GevSCPSFireTestPacket](#)  
Description: Sends a test packet.
- [GenApi::String](#) & [GevSecondURL](#)  
Description: The second choice of URL to the XML device description file.
- [GenApi::EnumerationT<GevSupportedOptionSelectorEnums>](#) & [GevSupportedOptionSelector](#)  
Description: Selects the GEV option to interrogate for existing support.
- [GenApi::Boolean](#) & [GevGVCPHeartbeatDisable](#)  
Description: Disables the GVCP heartbeat.
- [GenApi::Integer](#) & [GevMCRC](#)  
Description: Indicates the number of retries of the message channel.
- [GenApi::Boolean](#) & [GevSCPSBigEndian](#)  
Description: Endianess of multi-byte pixel data for this stream.
- [GenApi::Integer](#) & [GevNumberOfInterfaces](#)  
Description: Indicates the number of physical network interfaces supported by this device.
- [GenApi::Integer](#) & [TLParamsLocked](#)  
Description: Visibility:
- [GenApi::Integer](#) & [PayloadSize](#)  
Description: Provides the number of bytes transferred for each image or chunk on the stream channel.
- [GenApi::Integer](#) & [PacketResendRequestCount](#)  
Description: Counts the number of resend requests received from the host.
- [GenApi::Boolean](#) & [SharpeningEnable](#)  
Description:  
Enables/disables the sharpening feature.
- [GenApi::EnumerationT<BlackLevelSelectorEnums>](#) & [BlackLevelSelector](#)  
Description:  
Selects which black level to control.
- [GenApi::Boolean](#) & [GammaEnable](#)  
Description: Enables/disables gamma correction.
- [GenApi::Boolean](#) & [SharpeningAuto](#)  
Description:  
Enables/disables the auto sharpening feature.
- [GenApi::Boolean](#) & [BlackLevelClampingEnable](#)  
Description:  
Enable the black level auto clamping feature which performing dark current compensation.
- [GenApi::Float](#) & [BalanceRatio](#)  
Description:  
Controls the balance ratio of the selected color relative to green.
- [GenApi::EnumerationT<BalanceWhiteAutoEnums>](#) & [BalanceWhiteAuto](#)  
Description:  
White Balance compensates for color shifts caused by different lighting conditions.
- [GenApi::Float](#) & [SharpeningThreshold](#)  
Description:  
Controls the minimum intensity gradient change to invoke sharpening.
- [GenApi::EnumerationT<GainAutoEnums>](#) & [GainAuto](#)  
Description:  
Sets the automatic gain mode.
- [GenApi::Float](#) & [Sharpening](#)  
Description:  
Controls the amount to sharpen a signal.
- [GenApi::Float](#) & [Gain](#)  
Description:  
Controls the amplification of the video signal in dB.

- [GenApi::IEnumerationT < BalanceRatioSelectorEnums > & BalanceRatioSelector](#)  
 Description:  
*Selects a balance ratio to configure once a balance ratio control has been selected.*
- [GenApi::IEnumerationT < GainSelectorEnums > & GainSelector](#)  
 Description: *Selects which gain to control.*
- [GenApi::IFloat & BlackLevel](#)  
 Description:  
*Controls the offset of the video signal in percent.*
- [GenApi::Integer & BlackLevelRaw](#)  
 Description:  
*Controls the offset of the video signal in camera specific units.*
- [GenApi::IFloat & Gamma](#)  
 Description: *Controls the gamma correction of pixel intensity.*
- [GenApi::Integer & DefectTableIndex](#)  
 Description:  
*Controls the offset of the element to access in the defective pixel location table.*
- [GenApi::ICommand & DefectTableFactoryRestore](#)  
 Description: *Restores the Defective Pixel Table to its factory default state, which was calibrated during manufacturing.*
- [GenApi::Integer & DefectTableCoordinateY](#)  
 Description:  
*Returns the Y coordinate of the defective pixel at DefectTableIndex within the defective pixel table.*
- [GenApi::ICommand & DefectTableSave](#)  
 Description: *Saves the current defective pixel table non-volatile memory, so that it is preserved when the camera boots up.*
- [GenApi::IEnumerationT < DefectCorrectionModeEnums > & DefectCorrectionMode](#)  
 Description: *Controls the method used for replacing defective pixels.*
- [GenApi::Integer & DefectTableCoordinateX](#)  
 Description:  
*Returns the X coordinate of the defective pixel at DefectTableIndex within the defective pixel table.*
- [GenApi::Integer & DefectTablePixelCount](#)  
 Description:  
*The number of defective pixel locations in the current table.*
- [GenApi::IBoolean & DefectCorrectStaticEnable](#)  
 Description: *Enables/Disables table-based defective pixel correction.*
- [GenApi::ICommand & DefectTableApply](#)  
 Description: *Applies the current defect table, so that any changes made affect images captured by the camera.*
- [GenApi::IBoolean & UserSetFeatureEnable](#)  
 Description: *Whether or not the selected feature is saved to user sets.*
- [GenApi::ICommand & UserSetSave](#)  
 Description:  
*Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.*
- [GenApi::IEnumerationT < UserSetSelectorEnums > & UserSetSelector](#)  
 Description:  
*Selects the feature User Set to load, save or configure.*
- [GenApi::ICommand & UserSetLoad](#)  
 Description:  
*Loads the User Set specified by UserSetSelector to the device and makes it active.*
- [GenApi::IEnumerationT < UserSetDefaultEnums > & UserSetDefault](#)  
 Description:  
*Selects the feature User Set to load and make active by default when the device is restarted.*
- [GenApi::IEnumerationT < SerialPortBaudRateEnums > & SerialPortBaudRate](#)  
 Description: *This feature controls the baud rate used by the selected serial port.*
- [GenApi::Integer & SerialPortDataBits](#)  
 Description: *This feature controls the number of data bits used by the selected serial port.*



- [GenApi::IEnumerationT< SerialPortParityEnums > & SerialPortParity](#)  
*Description: This feature controls the parity used by the selected serial port.*
- [GenApi::Integer & SerialTransmitQueueMaxCharacterCount](#)  
*Description: >Returns the maximum number of characters in the serial port transmit queue.*
- [GenApi::Integer & SerialReceiveQueueCurrentCharacterCount](#)  
*Description: Returns the number of characters currently in the serial port receive queue.*
- [GenApi::IEnumerationT< SerialPortSelectorEnums > & SerialPortSelector](#)  
*Description: Selects which serial port of the device to control.*
- [GenApi::IEnumerationT< SerialPortStopBitsEnums > & SerialPortStopBits](#)  
*Description: This feature controls the number of stop bits used by the selected serial port.*
- [GenApi::ICommand & SerialReceiveQueueClear](#)  
*Description: This is a command that clears the device serial port receive queue.*
- [GenApi::Integer & SerialReceiveFramingErrorCount](#)  
*Description: Returns the number of framing errors that have occurred on the serial port.*
- [GenApi::Integer & SerialTransmitQueueCurrentCharacterCount](#)  
*Description: Returns the number of characters currently in the serial port transmit queue.*
- [GenApi::Integer & SerialReceiveParityErrorCount](#)  
*Description: Returns the number of parity errors that have occurred on the serial port.*
- [GenApi::IEnumerationT< SerialPortSourceEnums > & SerialPortSource](#)  
*Description: Specifies the physical input Line on which to receive serial data.*
- [GenApi::Integer & SerialReceiveQueueMaxCharacterCount](#)  
*Description: >Returns the maximum number of characters in the serial port receive queue.*
- [GenApi::Integer & SequencerSetStart](#)  
*Description: Sets the first sequencer set to be used.*
- [GenApi::IEnumerationT< SequencerModeEnums > & SequencerMode](#)  
*Description: Controls whether or not a sequencer is active.*
- [GenApi::IEnumerationT< SequencerConfigurationValidEnums > & SequencerConfigurationValid](#)  
*Description:*  
*Display whether the current sequencer configuration is valid to run.*
- [GenApi::IEnumerationT< SequencerSetValidEnums > & SequencerSetValid](#)  
*Description:*  
*Displays whether the currently selected sequencer set's register contents are valid to use.*
- [GenApi::Integer & SequencerSetSelector](#)  
*Description:*  
*Selects the sequencer set to which subsequent settings apply.*
- [GenApi::IEnumerationT< SequencerTriggerActivationEnums > & SequencerTriggerActivation](#)  
*Description:*  
*Specifies the activation mode of the sequencer trigger.*
- [GenApi::IEnumerationT< SequencerConfigurationModeEnums > & SequencerConfigurationMode](#)  
*Description:*  
*Controls whether or not a sequencer is in configuration mode.*
- [GenApi::ICommand & SequencerSetSave](#)  
*Description:*  
*Saves the current device configuration to the currently selected sequencer set.*
- [GenApi::IEnumerationT< SequencerTriggerSourceEnums > & SequencerTriggerSource](#)  
*Description:*  
*Specifies the internal signal or physical input line to use as the sequencer trigger source.*
- [GenApi::Integer & SequencerSetActive](#)  
*Description: Displays the currently active sequencer set.*
- [GenApi::Integer & SequencerSetNext](#)  
*Description: Specifies the next sequencer set.*
- [GenApi::ICommand & SequencerSetLoad](#)



Description:

*Loads currently selected sequencer to the current device configuration.*

- [GenApi::Integer](#) & [SequencerPathSelector](#)

Description:

*Selects branching path to be used for subsequent settings.*

- [GenApi::Boolean](#) & [SequencerFeatureEnable](#)

Description:

*Enables the selected feature and makes it active in all sequencer sets.*

- [GenApi::Integer](#) & [TransferBlockCount](#)

*Description: Specifies the number of data blocks (images) that the device should stream before stopping.*

- [GenApi::Command](#) & [TransferStart](#)

*Description: Starts the streaming of data blocks (images) out of the device.*

- [GenApi::Integer](#) & [TransferQueueMaxBlockCount](#)

*Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:*

- [GenApi::Integer](#) & [TransferQueueCurrentBlockCount](#)

*Description: Returns number of data blocks (images) currently in the transfer queue.*

- [GenApi::EnumerationT< TransferQueueModeEnums >](#) & [TransferQueueMode](#)

*Description: Specifies the operation mode of the transfer queue.*

- [GenApi::EnumerationT< TransferOperationModeEnums >](#) & [TransferOperationMode](#)

*Description: Selects the operation mode of the transfer.*

- [GenApi::Command](#) & [TransferStop](#)

*Description: Stops the streaming of data block (images).*

- [GenApi::Integer](#) & [TransferQueueOverflowCount](#)

*Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.*

- [GenApi::EnumerationT< TransferControlModeEnums >](#) & [TransferControlMode](#)

*Description: Selects the control method for the transfers.*

- [GenApi::Float](#) & [ChunkBlackLevel](#)

*Description: Returns the black level used to capture the image.*

- [GenApi::Integer](#) & [ChunkFrameID](#)

*Description: Returns the image frame ID.*

- [GenApi::String](#) & [ChunkSerialData](#)

*Description: Returns the serial data that was received.*

- [GenApi::Float](#) & [ChunkExposureTime](#)

*Description: Returns the exposure time used to capture the image.*

- [GenApi::Boolean](#) & [ChunkSerialReceiveOverflow](#)

*Description: Returns the status of the chunk serial receive overflow.*

- [GenApi::Integer](#) & [ChunkTimestamp](#)

*Description: Returns the Timestamp of the image.*

- [GenApi::Boolean](#) & [ChunkModeActive](#)

*Description: Activates the inclusion of Chunk data in the payload of the image.*

- [GenApi::Integer](#) & [ChunkExposureEndLineStatusAll](#)

*Description: Returns the status of all the I/O lines at the end of exposure event.*

- [GenApi::EnumerationT< ChunkGainSelectorEnums >](#) & [ChunkGainSelector](#)

*Description: Selects which gain to retrieve Visibility:*

- [GenApi::EnumerationT< ChunkSelectorEnums >](#) & [ChunkSelector](#)

*Description: Selects which chunk data to enable or disable.*

- [GenApi::EnumerationT< ChunkBlackLevelSelectorEnums >](#) & [ChunkBlackLevelSelector](#)

*Description: Selects which black level to retrieve Visibility:*

- [GenApi::Integer](#) & [ChunkWidth](#)

*Description: Returns the width of the image included in the payload.*

- [GenApi::Integer](#) & [ChunkImage](#)

- Description: Returns the image payload.*

  - [GenApi::Integer](#) & [ChunkHeight](#)

*Description: Returns the height of the image included in the payload.*
- [GenApi::EnumerationT< ChunkPixelFormatEnums >](#) & [ChunkPixelFormat](#)

*Description: Format of the pixel provided by the camera Visibility:*
- [GenApi::IFloat](#) & [ChunkGain](#)

*Description: Returns the gain used to capture the image.*
- [GenApi::Integer](#) & [ChunkSequencerSetActive](#)

*Description: Returns the index of the active set of the running sequencer included in the payload.*
- [GenApi::Integer](#) & [ChunkCRC](#)

*Description: Returns the CRC of the image payload.*
- [GenApi::Integer](#) & [ChunkOffsetX](#)

*Description: Returns the Offset X of the image included in the payload.*
- [GenApi::Integer](#) & [ChunkOffsetY](#)

*Description: Returns the Offset Y of the image included in the payload.*
- [GenApi::Boolean](#) & [ChunkEnable](#)

*Description: Enables the inclusion of the selected Chunk data in the payload of the image.*
- [GenApi::Integer](#) & [ChunkSerialDataLength](#)

*Description: Returns the length of the received serial data that was included in the payload.*
- [GenApi::Integer](#) & [FileAccessOffset](#)

*Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.*
- [GenApi::Integer](#) & [FileAccessLength](#)

*Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.*
- [GenApi::EnumerationT< FileOperationStatusEnums >](#) & [FileOperationStatus](#)

*Description: Represents the file operation execution status.*
- [GenApi::ICommand](#) & [FileOperationExecute](#)

*Description:*  
*This is a command that executes the selected file operation on the selected file.*
- [GenApi::EnumerationT< FileOpenModeEnums >](#) & [FileOpenMode](#)

*Description:*  
*The mode of the file when it is opened.*
- [GenApi::Integer](#) & [FileOperationResult](#)

*Description: Represents the file operation result.*
- [GenApi::EnumerationT< FileOperationSelectorEnums >](#) & [FileOperationSelector](#)

*Description:*  
*Sets operation to execute on the selected file when the execute command is given.*
- [GenApi::EnumerationT< FileSelectorEnums >](#) & [FileSelector](#)

*Description:*  
*Selects which file is being operated on.*
- [GenApi::Integer](#) & [FileSize](#)

*Description: Represents the size of the selected file in bytes.*
- [GenApi::EnumerationT< BinningSelectorEnums >](#) & [BinningSelector](#)

*Description:*  
*Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.*
- [GenApi::Integer](#) & [PixelDynamicRangeMin](#)

*Description: Minimum value that can be returned during the digitization process.*
- [GenApi::Integer](#) & [PixelDynamicRangeMax](#)

*Description: Maximum value that can be returned during the digitization process.*
- [GenApi::Integer](#) & [OffsetY](#)

*Description:*  
*Vertical offset from the origin to the ROI (in pixels).*
- [GenApi::Integer](#) & [BinningHorizontal](#)

- Description:
  - Number of horizontal photo-sensitive cells to combine together.
- [GenApi::Integer](#) & [Width](#)
  - Description:
  - Width of the image provided by the device (in pixels).
- [GenApi::EnumerationT< TestPatternGeneratorSelectorEnums >](#) & [TestPatternGeneratorSelector](#)
  - Description:
  - Selects which test pattern generator is controlled by the TestPattern feature.
- [GenApi::IFloat](#) & [CompressionRatio](#)
  - Description: Reports the ratio between the uncompressed image size and compressed image size.
- [GenApi::IBoolean](#) & [ReverseX](#)
  - Description: Horizontally flips the image sent by the device.
- [GenApi::IBoolean](#) & [ReverseY](#)
  - Description: Vertically flips the image sent by the device.
- [GenApi::EnumerationT< TestPatternEnums >](#) & [TestPattern](#)
  - Description:
  - Selects the type of test pattern that is generated by the device as image source.
- [GenApi::EnumerationT< PixelColorFilterEnums >](#) & [PixelColorFilter](#)
  - Description: Type of color filter that is applied to the image.
- [GenApi::Integer](#) & [WidthMax](#)
  - Description:
  - Maximum width of the image (in pixels).
- [GenApi::EnumerationT< AdcBitDepthEnums >](#) & [AdcBitDepth](#)
  - Description:
  - Selects which ADC bit depth to use.
- [GenApi::Integer](#) & [BinningVertical](#)
  - Description:
  - Number of vertical photo-sensitive cells to combine together.
- [GenApi::EnumerationT< DecimationHorizontalModeEnums >](#) & [DecimationHorizontalMode](#)
  - Description:
  - The mode used to reduce the horizontal resolution when DecimationHorizontal is used.
- [GenApi::EnumerationT< BinningVerticalModeEnums >](#) & [BinningVerticalMode](#)
  - Description: Visibility:
- [GenApi::Integer](#) & [OffsetX](#)
  - Description:
  - Horizontal offset from the origin to the ROI (in pixels).
- [GenApi::Integer](#) & [HeightMax](#)
  - Description: Maximum height of the image (in pixels).
- [GenApi::Integer](#) & [DecimationHorizontal](#)
  - Description:
  - Horizontal decimation of the image.
- [GenApi::EnumerationT< PixelSizeEnums >](#) & [PixelSize](#)
  - Description: Total size in bits of a pixel of the image.
- [GenApi::Integer](#) & [SensorHeight](#)
  - Description: Effective height of the sensor in pixels.
- [GenApi::EnumerationT< DecimationSelectorEnums >](#) & [DecimationSelector](#)
  - Description: Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.
- [GenApi::IBoolean](#) & [IspEnable](#)
  - Description:
  - Controls whether the image processing core is used for optional pixel format mode (i.e.
- [GenApi::IBoolean](#) & [AdaptiveCompressionEnable](#)
  - Description: Controls whether lossless compression adapts to the image content.
- [GenApi::EnumerationT< ImageCompressionModeEnums >](#) & [ImageCompressionMode](#)

- Description: Visibility:*

  - [GenApi::Integer](#) & [DecimationVertical](#)

*Description:*  
Vertical decimation of the image.
  - [GenApi::Integer](#) & [Height](#)

*Description:*  
Height of the image provided by the device (in pixels).
  - [GenApi::EnumerationT](#)< [BinningHorizontalModeEnums](#) > & [BinningHorizontalMode](#)

*Description: Visibility:*
  - [GenApi::EnumerationT](#)< [PixelFormatEnums](#) > & [PixelFormat](#)

*Description: Format of the pixel provided by the camera.*
  - [GenApi::Integer](#) & [SensorWidth](#)

*Description: Effective width of the sensor in pixels.*
  - [GenApi::EnumerationT](#)< [DecimationVerticalModeEnums](#) > & [DecimationVerticalMode](#)

*Description:*  
The mode used to reduce the vertical resolution when DecimationVertical is used.
  - [GenApi::ICommand](#) & [TestEventGenerate](#)

*Description: This command generates a test event and sends it to the host.*
  - [GenApi::ICommand](#) & [TriggerEventTest](#)

*Description: This command generates a test event and sends it to the host.*
  - [GenApi::Integer](#) & [GuiXmlManifestAddress](#)

*Description: Location of the GUI XML manifest table.*
  - [GenApi::Integer](#) & [Test0001](#)

*Description: For testing only.*
  - [GenApi::Boolean](#) & [V3\\_3Enable](#)

*Description: Internally generated 3.3V rail.*
  - [GenApi::EnumerationT](#)< [LineModeEnums](#) > & [LineMode](#)

*Description: Controls if the physical Line is used to Input or Output a signal.*
  - [GenApi::EnumerationT](#)< [LineSourceEnums](#) > & [LineSource](#)

*Description: Selects which internal acquisition or I/O source signal to output on the selected line.*
  - [GenApi::EnumerationT](#)< [LineInputFilterSelectorEnums](#) > & [LineInputFilterSelector](#)

*Description: Selects the kind of input filter to configure: Deglitch or Debounce.*
  - [GenApi::Boolean](#) & [UserOutputValue](#)

*Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).*
  - [GenApi::Integer](#) & [UserOutputValueAll](#)

*Description: Returns the current status of all the user output status bits in a hexadecimal representation (UserOutput 0 status corresponds to bit 0, UserOutput 1 status with bit 1, etc).*
  - [GenApi::EnumerationT](#)< [UserOutputSelectorEnums](#) > & [UserOutputSelector](#)

*Description: Selects which bit of the User Output register is set by UserOutputValue.*
  - [GenApi::Boolean](#) & [LineStatus](#)

*Description: Returns the current status of the selected input or output Line Visibility:*
  - [GenApi::EnumerationT](#)< [LineFormatEnums](#) > & [LineFormat](#)

*Description: Displays the current electrical format of the selected physical input or output Line.*
  - [GenApi::Integer](#) & [LineStatusAll](#)

*Description: Returns the current status of all the line status bits in a hexadecimal representation (Line 0 status corresponds to bit 0, Line 1 status with bit 1, etc).*
  - [GenApi::EnumerationT](#)< [LineSelectorEnums](#) > & [LineSelector](#)

*Description: Selects the physical line (or pin) of the external device connector to configure Visibility:*
  - [GenApi::EnumerationT](#)< [ExposureActiveModeEnums](#) > & [ExposureActiveMode](#)

*Description: Control sensor active exposure mode.*
  - [GenApi::Boolean](#) & [LineInverter](#)

*Description: Controls the inversion of the signal of the selected input or output line.*

- [GenApi::IFloat](#) & [LineFilterWidth](#)  
Description: Filter width in microseconds for the selected line and filter combination Visibility:
- [GenApi::IEnumerationT< CounterTriggerActivationEnums >](#) & [CounterTriggerActivation](#)  
Description: Selects the activation mode of the trigger to start the Counter.
- [GenApi::Integer](#) & [CounterValue](#)  
Description: Current counter value Visibility:
- [GenApi::IEnumerationT< CounterSelectorEnums >](#) & [CounterSelector](#)  
Description: Selects which counter to configure Visibility:
- [GenApi::Integer](#) & [CounterValueAtReset](#)  
Description: Value of the selected Counter when it was reset by a trigger.
- [GenApi::IEnumerationT< CounterStatusEnums >](#) & [CounterStatus](#)  
Description: Returns the current status of the Counter.
- [GenApi::IEnumerationT< CounterTriggerSourceEnums >](#) & [CounterTriggerSource](#)  
Description: Selects the source of the trigger to start the counter Visibility:
- [GenApi::Integer](#) & [CounterDelay](#)  
Description: Sets the delay (or number of events) before the CounterStart event is generated.
- [GenApi::IEnumerationT< CounterResetSourceEnums >](#) & [CounterResetSource](#)  
Description: Selects the signal that will be the source to reset the Counter.
- [GenApi::IEnumerationT< CounterEventSourceEnums >](#) & [CounterEventSource](#)  
Description: Selects the event that will increment the counter Visibility:
- [GenApi::IEnumerationT< CounterEventActivationEnums >](#) & [CounterEventActivation](#)  
Description: Selects the activation mode of the event to increment the Counter.
- [GenApi::Integer](#) & [CounterDuration](#)  
Description: Sets the duration (or number of events) before the CounterEnd event is generated.
- [GenApi::IEnumerationT< CounterResetActivationEnums >](#) & [CounterResetActivation](#)  
Description: Selects the Activation mode of the Counter Reset Source signal.
- [GenApi::IEnumerationT< DeviceTypeEnums >](#) & [DeviceType](#)  
Description: Returns the device type.
- [GenApi::IString](#) & [DeviceFamilyName](#)  
Description: Identifier of the product family of the device.
- [GenApi::Integer](#) & [DeviceSFNCVersionMajor](#)  
Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceSFNCVersionMinor](#)  
Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceSFNCVersionSubMinor](#)  
Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceManifestEntrySelector](#)  
Description: Selects the manifest entry to reference.
- [GenApi::Integer](#) & [DeviceManifestXMLMajorVersion](#)  
Description: Indicates the major version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestXMLMinorVersion](#)  
Description: Indicates the minor version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestXMLSubMinorVersion](#)  
Description: Indicates the subminor version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestSchemaMajorVersion](#)  
Description: Indicates the major version number of the schema file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestSchemaMinorVersion](#)  
Description: Indicates the minor version number of the schema file of the selected manifest entry.

- [GenApi::IString](#) & [DeviceManifestPrimaryURL](#)  
Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.
- [GenApi::IString](#) & [DeviceManifestSecondaryURL](#)  
Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceTLVersionSubMinor](#)  
Description: Sub minor version of the Transport Layer of the device.
- [GenApi::Integer](#) & [DeviceGenCPVersionMajor](#)  
Description: Major version of the GenCP protocol supported by the device.
- [GenApi::Integer](#) & [DeviceGenCPVersionMinor](#)  
Description: Minor version of the GenCP protocol supported by the device.
- [GenApi::Integer](#) & [DeviceConnectionSelector](#)  
Description: Selects which Connection of the device to control.
- [GenApi::Integer](#) & [DeviceConnectionSpeed](#)  
Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.
- [GenApi::EnumerationT](#) < [DeviceConnectionStatusEnums](#) > & [DeviceConnectionStatus](#)  
Description: Indicates the status of the specified Connection.
- [GenApi::Integer](#) & [DeviceLinkSelector](#)  
Description: Selects which Link of the device to control.
- [GenApi::EnumerationT](#) < [DeviceLinkThroughputLimitModeEnums](#) > & [DeviceLinkThroughputLimitMode](#)  
Description: Controls if the DeviceLinkThroughputLimit is active.
- [GenApi::Integer](#) & [DeviceLinkConnectionCount](#)  
Description: Returns the number of physical connection of the device used by a particular Link.
- [GenApi::EnumerationT](#) < [DeviceLinkHeartbeatModeEnums](#) > & [DeviceLinkHeartbeatMode](#)  
Description: Activate or deactivate the Link's heartbeat.
- [GenApi::IFloat](#) & [DeviceLinkHeartbeatTimeout](#)  
Description: Controls the current heartbeat timeout of the specific Link.
- [GenApi::IFloat](#) & [DeviceLinkCommandTimeout](#)  
Description: Indicates the command timeout of the specified Link.
- [GenApi::Integer](#) & [DeviceStreamChannelSelector](#)  
Description: Selects the stream channel to control.
- [GenApi::EnumerationT](#) < [DeviceStreamChannelTypeEnums](#) > & [DeviceStreamChannelType](#)  
Description: Reports the type of the stream channel.
- [GenApi::Integer](#) & [DeviceStreamChannelLink](#)  
Description: Index of device's Link to use for streaming the specified stream channel.
- [GenApi::EnumerationT](#) < [DeviceStreamChannelEndiannessEnums](#) > & [DeviceStreamChannelEndianness](#)  
Description: Endianness of multi-byte pixel data for this stream.
- [GenApi::Integer](#) & [DeviceStreamChannelPacketSize](#)  
Description: Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.
- [GenApi::ICommand](#) & [DeviceFeaturePersistenceStart](#)  
Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.
- [GenApi::ICommand](#) & [DeviceFeaturePersistenceEnd](#)  
Description: Indicate to the device the end of feature persistence.
- [GenApi::ICommand](#) & [DeviceRegistersStreamingStart](#)  
Description: Prepare the device for registers streaming without checking for consistency.
- [GenApi::ICommand](#) & [DeviceRegistersStreamingEnd](#)  
Description: Announce the end of registers streaming.
- [GenApi::ICommand](#) & [DeviceRegistersCheck](#)  
Description: Perform the validation of the current register set for consistency.
- [GenApi::IBoolean](#) & [DeviceRegistersValid](#)  
Description: Returns if the current register set is valid and consistent.

- [GenApi::IEnumerationT< DeviceClockSelectorEnums > & DeviceClockSelector](#)  
Description: Selects the clock frequency to access from the device.
- [GenApi::IFloat & DeviceClockFrequency](#)  
Description: Returns the frequency of the selected Clock.
- [GenApi::IEnumerationT< DeviceSerialPortSelectorEnums > & DeviceSerialPortSelector](#)  
Description: Selects which serial port of the device to control.
- [GenApi::IEnumerationT< DeviceSerialPortBaudRateEnums > & DeviceSerialPortBaudRate](#)  
Description: This feature controls the baud rate used by the selected serial port.
- [GenApi::Integer & Timestamp](#)  
Description: Reports the current value of the device timestamp counter.
- [GenApi::IEnumerationT< SensorTapsEnums > & SensorTaps](#)  
Description: Number of taps of the camera sensor.
- [GenApi::IEnumerationT< SensorDigitizationTapsEnums > & SensorDigitizationTaps](#)  
Description: Number of digitized samples outputted simultaneously by the camera A/D conversion stage.
- [GenApi::IEnumerationT< RegionSelectorEnums > & RegionSelector](#)  
Description: Selects the Region of interest to control.
- [GenApi::IEnumerationT< RegionModeEnums > & RegionMode](#)  
Description: Controls if the selected Region of interest is active and streaming.
- [GenApi::IEnumerationT< RegionDestinationEnums > & RegionDestination](#)  
Description: Control the destination of the selected region.
- [GenApi::IEnumerationT< ImageComponentSelectorEnums > & ImageComponentSelector](#)  
Description: Selects a component to activate data streaming from.
- [GenApi::IBoolean & ImageComponentEnable](#)  
Description: Controls if the selected component streaming is active.
- [GenApi::Integer & LinePitch](#)  
Description: Total number of bytes between 2 successive lines.
- [GenApi::IEnumerationT< PixelFormatInfoSelectorEnums > & PixelFormatInfoSelector](#)  
Description: Select the pixel format for which the information will be returned.
- [GenApi::Integer & PixelFormatInfoID](#)  
Description: Returns the value used by the streaming channels to identify the selected pixel format.
- [GenApi::IEnumerationT< DeinterlacingEnums > & Deinterlacing](#)  
Description: Controls how the device performs de-interlacing.
- [GenApi::IEnumerationT< ImageCompressionRateOptionEnums > & ImageCompressionRateOption](#)  
Description: Two rate controlling options are offered: fixed bit rate or fixed quality.
- [GenApi::Integer & ImageCompressionQuality](#)  
Description: Control the quality of the produced compressed stream.
- [GenApi::IFloat & ImageCompressionBitrate](#)  
Description: Control the rate of the produced compressed stream.
- [GenApi::IEnumerationT< ImageCompressionJPEGFormatOptionEnums > & ImageCompressionJPEGFormatOption](#)  
Description: When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.
- [GenApi::ICommand & AcquisitionAbort](#)  
Description: Aborts the Acquisition immediately.
- [GenApi::ICommand & AcquisitionArm](#)  
Description: Arms the device before an AcquisitionStart command.
- [GenApi::IEnumerationT< AcquisitionStatusSelectorEnums > & AcquisitionStatusSelector](#)  
Description: Selects the internal acquisition signal to read using AcquisitionStatus.
- [GenApi::IBoolean & AcquisitionStatus](#)  
Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.
- [GenApi::Integer & TriggerDivider](#)



- Description: Specifies a division factor for the incoming trigger pulses.*

  - [GenApi::Integer](#) & [TriggerMultiplier](#)
- Description: Specifies a multiplication factor for the incoming trigger pulses.*

  - [GenApi::EnumerationT< ExposureTimeModeEnums >](#) & [ExposureTimeMode](#)
- Description: Sets the configuration mode of the ExposureTime feature.*

  - [GenApi::EnumerationT< ExposureTimeSelectorEnums >](#) & [ExposureTimeSelector](#)
- Description: Selects which exposure time is controlled by the ExposureTime feature.*

  - [GenApi::EnumerationT< GainAutoBalanceEnums >](#) & [GainAutoBalance](#)
- Description: Sets the mode for automatic gain balancing between the sensor color channels or taps.*

  - [GenApi::EnumerationT< BlackLevelAutoEnums >](#) & [BlackLevelAuto](#)
- Description: Controls the mode for automatic black level adjustment.*

  - [GenApi::EnumerationT< BlackLevelAutoBalanceEnums >](#) & [BlackLevelAutoBalance](#)
- Description: Controls the mode for automatic black level balancing between the sensor color channels or taps.*

  - [GenApi::EnumerationT< WhiteClipSelectorEnums >](#) & [WhiteClipSelector](#)
- Description: Selects which White Clip to control.*

  - [GenApi::IFloat](#) & [WhiteClip](#)
- Description: Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.*

  - [GenApi::IRegister](#) & [LUTValueAll](#)
- Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.*

  - [GenApi::Integer](#) & [UserOutputValueAllMask](#)
- Description: Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.*

  - [GenApi::ICommand](#) & [CounterReset](#)
- Description: Does a software reset of the selected Counter and starts it.*

  - [GenApi::EnumerationT< TimerSelectorEnums >](#) & [TimerSelector](#)
- Description: Selects which Timer to configure.*

  - [GenApi::IFloat](#) & [TimerDuration](#)
- Description: Sets the duration (in microseconds) of the Timer pulse.*

  - [GenApi::IFloat](#) & [TimerDelay](#)
- Description: Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer.*

  - [GenApi::ICommand](#) & [TimerReset](#)
- Description: Does a software reset of the selected timer and starts it.*

  - [GenApi::IFloat](#) & [TimerValue](#)
- Description: Reads or writes the current value (in microseconds) of the selected Timer.*

  - [GenApi::EnumerationT< TimerStatusEnums >](#) & [TimerStatus](#)
- Description: Returns the current status of the Timer.*

  - [GenApi::EnumerationT< TimerTriggerSourceEnums >](#) & [TimerTriggerSource](#)
- Description: Selects the source of the trigger to start the Timer.*

  - [GenApi::EnumerationT< TimerTriggerActivationEnums >](#) & [TimerTriggerActivation](#)
- Description: Selects the activation mode of the trigger to start the Timer.*

  - [GenApi::EnumerationT< EncoderSelectorEnums >](#) & [EncoderSelector](#)
- Description: Selects which Encoder to configure.*

  - [GenApi::EnumerationT< EncoderSourceAEnums >](#) & [EncoderSourceA](#)
- Description: Selects the signal which will be the source of the A input of the Encoder.*

  - [GenApi::EnumerationT< EncoderSourceBEnums >](#) & [EncoderSourceB](#)
- Description: Selects the signal which will be the source of the B input of the Encoder.*

  - [GenApi::EnumerationT< EncoderModeEnums >](#) & [EncoderMode](#)
- Description: Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.*



- [GenApi::Integer](#) & [EncoderDivider](#)  
*Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.*
- [GenApi::EnumerationT< EncoderOutputModeEnums >](#) & [EncoderOutputMode](#)  
*Description: Selects the conditions for the Encoder interface to generate a valid Encoder output signal.*
- [GenApi::EnumerationT< EncoderStatusEnums >](#) & [EncoderStatus](#)  
*Description: Returns the motion status of the encoder.*
- [GenApi::IFloat](#) & [EncoderTimeout](#)  
*Description: Sets the maximum time interval between encoder counter increments before the status turns to static.*
- [GenApi::EnumerationT< EncoderResetSourceEnums >](#) & [EncoderResetSource](#)  
*Description: Selects the signals that will be the source to reset the Encoder.*
- [GenApi::EnumerationT< EncoderResetActivationEnums >](#) & [EncoderResetActivation](#)  
*Description: Selects the Activation mode of the Encoder Reset Source signal.*
- [GenApi::ICommand](#) & [EncoderReset](#)  
*Description: Does a software reset of the selected Encoder and starts it.*
- [GenApi::Integer](#) & [EncoderValue](#)  
*Description: Reads or writes the current value of the position counter of the selected Encoder.*
- [GenApi::Integer](#) & [EncoderValueAtReset](#)  
*Description: Reads the value of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.*
- [GenApi::EnumerationT< SoftwareSignalSelectorEnums >](#) & [SoftwareSignalSelector](#)  
*Description: Selects which Software Signal features to control.*
- [GenApi::ICommand](#) & [SoftwareSignalPulse](#)  
*Description: Generates a pulse signal that can be used as a software trigger.*
- [GenApi::EnumerationT< ActionUnconditionalModeEnums >](#) & [ActionUnconditionalMode](#)  
*Description: Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.*
- [GenApi::Integer](#) & [ActionDeviceKey](#)  
*Description: Provides the device key that allows the device to check the validity of action commands.*
- [GenApi::Integer](#) & [ActionQueueSize](#)  
*Description: Indicates the size of the scheduled action commands queue.*
- [GenApi::Integer](#) & [ActionSelector](#)  
*Description: Selects to which Action Signal further Action settings apply.*
- [GenApi::Integer](#) & [ActionGroupMask](#)  
*Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.*
- [GenApi::Integer](#) & [ActionGroupKey](#)  
*Description: Provides the key that the device will use to validate the action on reception of the action protocol message.*
- [GenApi::Integer](#) & [EventAcquisitionTrigger](#)  
*Description: Returns the unique Identifier of the Acquisition Trigger type of [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionTriggerTimestamp](#)  
*Description: Returns the Timestamp of the Acquisition Trigger [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionTriggerFrameID](#)  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionStart](#)  
*Description: Returns the unique Identifier of the Acquisition Start type of [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionStartTimestamp](#)  
*Description: Returns the Timestamp of the Acquisition Start [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionStartFrameID](#)  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionEnd](#)

- Description: Returns the unique Identifier of the Acquisition End type of [Event](#).*

  - [GenApi::Integer](#) & [EventAcquisitionEndTimestamp](#)

*Description: Returns the Timestamp of the Acquisition End [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionEndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionTransferStart](#)

*Description: Returns the unique Identifier of the Acquisition Transfer Start type of [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionTransferStartTimestamp](#)

*Description: Returns the Timestamp of the Acquisition Transfer Start [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionTransferStartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionTransferEnd](#)

*Description: Returns the unique Identifier of the Acquisition Transfer End type of [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionTransferEndTimestamp](#)

*Description: Returns the Timestamp of the Acquisition Transfer End [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionTransferEndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionError](#)

*Description: Returns the unique Identifier of the Acquisition Error type of [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionErrorTimestamp](#)

*Description: Returns the Timestamp of the Acquisition Error [Event](#).*
- [GenApi::Integer](#) & [EventAcquisitionErrorFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error [Event](#).*
- [GenApi::Integer](#) & [EventFrameTrigger](#)

*Description: Returns the unique Identifier of the FrameTrigger type of [Event](#).*
- [GenApi::Integer](#) & [EventFrameTriggerTimestamp](#)

*Description: Returns the Timestamp of the FrameTrigger [Event](#).*
- [GenApi::Integer](#) & [EventFrameTriggerFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger [Event](#).*
- [GenApi::Integer](#) & [EventFrameStart](#)

*Description: Returns the unique Identifier of the Frame Start type of [Event](#).*
- [GenApi::Integer](#) & [EventFrameStartTimestamp](#)

*Description: Returns the Timestamp of the Frame Start [Event](#).*
- [GenApi::Integer](#) & [EventFrameStartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start [Event](#).*
- [GenApi::Integer](#) & [EventFrameEnd](#)

*Description: Returns the unique Identifier of the Frame End type of [Event](#).*
- [GenApi::Integer](#) & [EventFrameEndTimestamp](#)

*Description: Returns the Timestamp of the Frame End [Event](#).*
- [GenApi::Integer](#) & [EventFrameEndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End [Event](#).*
- [GenApi::Integer](#) & [EventFrameBurstStart](#)

*Description: Returns the unique Identifier of the Frame Burst Start type of [Event](#).*
- [GenApi::Integer](#) & [EventFrameBurstStartTimestamp](#)

*Description: Returns the Timestamp of the Frame Burst Start [Event](#).*
- [GenApi::Integer](#) & [EventFrameBurstStartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start [Event](#).*
- [GenApi::Integer](#) & [EventFrameBurstEnd](#)

*Description: Returns the unique Identifier of the Frame Burst End type of [Event](#).*
- [GenApi::Integer](#) & [EventFrameBurstEndTimestamp](#)

*Description: Returns the Timestamp of the Frame Burst End [Event](#).*

- [GenApi::Integer](#) & [EventFrameBurstEndFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferStart](#)  
Description: Returns the unique Identifier of the Frame Transfer Start type of [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferStartTimestamp](#)  
Description: Returns the Timestamp of the Frame Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferStartFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferEnd](#)  
Description: Returns the unique Identifier of the Frame Transfer End type of [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferEndTimestamp](#)  
Description: Returns the Timestamp of the Frame Transfer End [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferEndFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End [Event](#).
- [GenApi::Integer](#) & [EventExposureStart](#)  
Description: Returns the unique Identifier of the Exposure Start type of [Event](#).
- [GenApi::Integer](#) & [EventExposureStartTimestamp](#)  
Description: Returns the Timestamp of the Exposure Start [Event](#).
- [GenApi::Integer](#) & [EventExposureStartFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferStart](#)  
Description: Returns the unique Identifier of the Stream 0 Transfer Start type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferStartTimestamp](#)  
Description: Returns the Timestamp of the Stream 0 Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferStartFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferEnd](#)  
Description: Returns the unique Identifier of the Stream 0 Transfer End type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferEndTimestamp](#)  
Description: Returns the Timestamp of the Stream 0 Transfer End [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferEndFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferPause](#)  
Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferPauseTimestamp](#)  
Description: Returns the Timestamp of the Stream 0 Transfer Pause [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferPauseFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferResume](#)  
Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferResumeTimestamp](#)  
Description: Returns the Timestamp of the Stream 0 Transfer Resume [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferResumeFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferBlockStart](#)  
Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferBlockStartTimestamp](#)  
Description: Returns the Timestamp of the Stream 0 Transfer Block Start [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferBlockStartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start Event.*

- [GenApi::Integer](#) & [EventStream0TransferBlockEnd](#)

*Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.*

- [GenApi::Integer](#) & [EventStream0TransferBlockEndTimestamp](#)

*Description: Returns the Timestamp of the Stream 0 Transfer Block End Event.*

- [GenApi::Integer](#) & [EventStream0TransferBlockEndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End Event.*

- [GenApi::Integer](#) & [EventStream0TransferBlockTrigger](#)

*Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.*

- [GenApi::Integer](#) & [EventStream0TransferBlockTriggerTimestamp](#)

*Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.*

- [GenApi::Integer](#) & [EventStream0TransferBlockTriggerFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger Event.*

- [GenApi::Integer](#) & [EventStream0TransferBurstStart](#)

*Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.*

- [GenApi::Integer](#) & [EventStream0TransferBurstStartTimestamp](#)

*Description: Returns the Timestamp of the Stream 0 Transfer Burst Start Event.*

- [GenApi::Integer](#) & [EventStream0TransferBurstStartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start Event.*

- [GenApi::Integer](#) & [EventStream0TransferBurstEnd](#)

*Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.*

- [GenApi::Integer](#) & [EventStream0TransferBurstEndTimestamp](#)

*Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.*

- [GenApi::Integer](#) & [EventStream0TransferBurstEndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End Event.*

- [GenApi::Integer](#) & [EventStream0TransferOverflow](#)

*Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.*

- [GenApi::Integer](#) & [EventStream0TransferOverflowTimestamp](#)

*Description: Returns the Timestamp of the Stream 0 Transfer Overflow Event.*

- [GenApi::Integer](#) & [EventStream0TransferOverflowFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.*

- [GenApi::Integer](#) & [EventSequencerSetChange](#)

*Description: Returns the unique Identifier of the Sequencer Set Change type of Event.*

- [GenApi::Integer](#) & [EventSequencerSetChangeTimestamp](#)

*Description: Returns the Timestamp of the Sequencer Set Change Event.*

- [GenApi::Integer](#) & [EventSequencerSetChangeFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.*

- [GenApi::Integer](#) & [EventCounter0Start](#)

*Description: Returns the unique Identifier of the Counter 0 Start type of Event.*

- [GenApi::Integer](#) & [EventCounter0StartTimestamp](#)

*Description: Returns the Timestamp of the Counter 0 Start Event.*

- [GenApi::Integer](#) & [EventCounter0StartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.*

- [GenApi::Integer](#) & [EventCounter1Start](#)

*Description: Returns the unique Identifier of the Counter 1 Start type of Event.*

- [GenApi::Integer](#) & [EventCounter1StartTimestamp](#)

- Description: Returns the Timestamp of the Counter 1 Start [Event](#).*

  - [GenApi::Integer](#) & [EventCounter1StartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start [Event](#).*

- [GenApi::Integer](#) & [EventCounter0End](#)

*Description: Returns the unique Identifier of the Counter 0 End type of [Event](#).*

- [GenApi::Integer](#) & [EventCounter0EndTimestamp](#)

*Description: Returns the Timestamp of the Counter 0 End [Event](#).*

- [GenApi::Integer](#) & [EventCounter0EndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End [Event](#).*

- [GenApi::Integer](#) & [EventCounter1End](#)

*Description: Returns the unique Identifier of the Counter 1 End type of [Event](#).*

- [GenApi::Integer](#) & [EventCounter1EndTimestamp](#)

*Description: Returns the Timestamp of the Counter 1 End [Event](#).*

- [GenApi::Integer](#) & [EventCounter1EndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End [Event](#).*

- [GenApi::Integer](#) & [EventTimer0Start](#)

*Description: Returns the unique Identifier of the Timer 0 Start type of [Event](#).*

- [GenApi::Integer](#) & [EventTimer0StartTimestamp](#)

*Description: Returns the Timestamp of the Timer 0 Start [Event](#).*

- [GenApi::Integer](#) & [EventTimer0StartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start [Event](#).*

- [GenApi::Integer](#) & [EventTimer1Start](#)

*Description: Returns the unique Identifier of the Timer 1 Start type of [Event](#).*

- [GenApi::Integer](#) & [EventTimer1StartTimestamp](#)

*Description: Returns the Timestamp of the Timer 1 Start [Event](#).*

- [GenApi::Integer](#) & [EventTimer1StartFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start [Event](#).*

- [GenApi::Integer](#) & [EventTimer0End](#)

*Description: Returns the unique Identifier of the Timer 0 End type of [Event](#).*

- [GenApi::Integer](#) & [EventTimer0EndTimestamp](#)

*Description: Returns the Timestamp of the Timer 0 End [Event](#).*

- [GenApi::Integer](#) & [EventTimer0EndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End [Event](#).*

- [GenApi::Integer](#) & [EventTimer1End](#)

*Description: Returns the unique Identifier of the Timer 1 End type of [Event](#).*

- [GenApi::Integer](#) & [EventTimer1EndTimestamp](#)

*Description: Returns the Timestamp of the Timer 1 End [Event](#).*

- [GenApi::Integer](#) & [EventTimer1EndFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End [Event](#).*

- [GenApi::Integer](#) & [EventEncoder0Stopped](#)

*Description: Returns the unique Identifier of the Encoder 0 Stopped type of [Event](#).*

- [GenApi::Integer](#) & [EventEncoder0StoppedTimestamp](#)

*Description: Returns the Timestamp of the Encoder 0 Stopped [Event](#).*

- [GenApi::Integer](#) & [EventEncoder0StoppedFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped [Event](#).*

- [GenApi::Integer](#) & [EventEncoder1Stopped](#)

*Description: Returns the unique Identifier of the Encoder 1 Stopped type of [Event](#).*

- [GenApi::Integer](#) & [EventEncoder1StoppedTimestamp](#)

*Description: Returns the Timestamp of the Encoder 1 Stopped [Event](#).*

- [GenApi::Integer](#) & [EventEncoder1StoppedFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped [Event](#).*

- [GenApi::Integer](#) & [EventEncoder0Restarted](#)  
Description: Returns the unique Identifier of the Encoder 0 Restarted type of [Event](#).
- [GenApi::Integer](#) & [EventEncoder0RestartedTimestamp](#)  
Description: Returns the Timestamp of the Encoder 0 Restarted [Event](#).
- [GenApi::Integer](#) & [EventEncoder0RestartedFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted [Event](#).
- [GenApi::Integer](#) & [EventEncoder1Restarted](#)  
Description: Returns the unique Identifier of the Encoder 1 Restarted type of [Event](#).
- [GenApi::Integer](#) & [EventEncoder1RestartedTimestamp](#)  
Description: Returns the Timestamp of the Encoder 1 Restarted [Event](#).
- [GenApi::Integer](#) & [EventEncoder1RestartedFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted [Event](#).
- [GenApi::Integer](#) & [EventLine0RisingEdge](#)  
Description: Returns the unique Identifier of the Line 0 Rising Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine0RisingEdgeTimestamp](#)  
Description: Returns the Timestamp of the Line 0 Rising Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0RisingEdgeFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1RisingEdge](#)  
Description: Returns the unique Identifier of the Line 1 Rising Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine1RisingEdgeTimestamp](#)  
Description: Returns the Timestamp of the Line 1 Rising Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1RisingEdgeFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0FallingEdge](#)  
Description: Returns the unique Identifier of the Line 0 Falling Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine0FallingEdgeTimestamp](#)  
Description: Returns the Timestamp of the Line 0 Falling Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0FallingEdgeFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1FallingEdge](#)  
Description: Returns the unique Identifier of the Line 1 Falling Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine1FallingEdgeTimestamp](#)  
Description: Returns the Timestamp of the Line 1 Falling Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1FallingEdgeFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0AnyEdge](#)  
Description: Returns the unique Identifier of the Line 0 Any Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine0AnyEdgeTimestamp](#)  
Description: Returns the Timestamp of the Line 0 Any Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0AnyEdgeFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1AnyEdge](#)  
Description: Returns the unique Identifier of the Line 1 Any Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine1AnyEdgeTimestamp](#)  
Description: Returns the Timestamp of the Line 1 Any Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1AnyEdgeFrameID](#)  
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge [Event](#).
- [GenApi::Integer](#) & [EventLinkTrigger0](#)  
Description: Returns the unique Identifier of the Link Trigger 0 type of [Event](#).
- [GenApi::Integer](#) & [EventLinkTrigger0Timestamp](#)



- Description: Returns the Timestamp of the Link Trigger 0 [Event](#).*

  - [GenApi::Integer](#) & [EventLinkTrigger0FrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 [Event](#).*
- [GenApi::Integer](#) & [EventLinkTrigger1](#)

*Description: Returns the unique Identifier of the Link Trigger 1 type of [Event](#).*
- [GenApi::Integer](#) & [EventLinkTrigger1Timestamp](#)

*Description: Returns the Timestamp of the Link Trigger 1 [Event](#).*
- [GenApi::Integer](#) & [EventLinkTrigger1FrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 [Event](#).*
- [GenApi::Integer](#) & [EventActionLate](#)

*Description: Returns the unique Identifier of the Action Late type of [Event](#).*
- [GenApi::Integer](#) & [EventActionLateTimestamp](#)

*Description: Returns the Timestamp of the Action Late [Event](#).*
- [GenApi::Integer](#) & [EventActionLateFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late [Event](#).*
- [GenApi::Integer](#) & [EventLinkSpeedChange](#)

*Description: Returns the unique Identifier of the Link Speed Change type of [Event](#).*
- [GenApi::Integer](#) & [EventLinkSpeedChangeTimestamp](#)

*Description: Returns the Timestamp of the Link Speed Change [Event](#).*
- [GenApi::Integer](#) & [EventLinkSpeedChangeFrameID](#)

*Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change [Event](#).*
- [GenApi::IRegister](#) & [FileAccessBuffer](#)

*Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.*
- [GenApi::Integer](#) & [SourceCount](#)

*Description: Controls or returns the number of sources supported by the device.*
- [GenApi::IEnumerationT](#) < [SourceSelectorEnums](#) > & [SourceSelector](#)

*Description: Selects the source to control.*
- [GenApi::IEnumerationT](#) < [TransferSelectorEnums](#) > & [TransferSelector](#)

*Description: Selects which stream transfers are currently controlled by the selected Transfer features.*
- [GenApi::Integer](#) & [TransferBurstCount](#)

*Description: Number of Block(s) to transfer for each TransferBurstStart trigger.*
- [GenApi::ICommand](#) & [TransferAbort](#)

*Description: Aborts immediately the streaming of data block(s).*
- [GenApi::ICommand](#) & [TransferPause](#)

*Description: Pauses the streaming of data Block(s).*
- [GenApi::ICommand](#) & [TransferResume](#)

*Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.*
- [GenApi::IEnumerationT](#) < [TransferTriggerSelectorEnums](#) > & [TransferTriggerSelector](#)

*Description: Selects the type of transfer trigger to configure.*
- [GenApi::IEnumerationT](#) < [TransferTriggerModeEnums](#) > & [TransferTriggerMode](#)

*Description: Controls if the selected trigger is active.*
- [GenApi::IEnumerationT](#) < [TransferTriggerSourceEnums](#) > & [TransferTriggerSource](#)

*Description: Specifies the signal to use as the trigger source for transfers.*
- [GenApi::IEnumerationT](#) < [TransferTriggerActivationEnums](#) > & [TransferTriggerActivation](#)

*Description: Specifies the activation mode of the transfer control trigger.*
- [GenApi::IEnumerationT](#) < [TransferStatusSelectorEnums](#) > & [TransferStatusSelector](#)

*Description: Selects which status of the transfer module to read.*
- [GenApi::IBoolean](#) & [TransferStatus](#)

*Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.*
- [GenApi::IEnumerationT](#) < [TransferComponentSelectorEnums](#) > & [TransferComponentSelector](#)

- Description: Selects the color component for the control of the TransferStreamChannel feature.*

  - [GenApi::Integer](#) & [TransferStreamChannel](#)
- Description: Selects the streaming channel that will be used to transfer the selected stream of data.*

  - [GenApi::EnumerationT< Scan3dDistanceUnitEnums >](#) & [Scan3dDistanceUnit](#)
- Description: Specifies the unit used when delivering calibrated distance data.*

  - [GenApi::EnumerationT< Scan3dCoordinateSystemEnums >](#) & [Scan3dCoordinateSystem](#)
- Description: Specifies the Coordinate system to use for the device.*

  - [GenApi::EnumerationT< Scan3dOutputModeEnums >](#) & [Scan3dOutputMode](#)
- Description: Controls the Calibration and data organization of the device, naming the coordinates transmitted.*

  - [GenApi::EnumerationT< Scan3dCoordinateSystemReferenceEnums >](#) & [Scan3dCoordinateSystemReference](#)
- Description: Defines coordinate system reference location.*

  - [GenApi::EnumerationT< Scan3dCoordinateSelectorEnums >](#) & [Scan3dCoordinateSelector](#)
- Description: Selects the individual coordinates in the vectors for 3D information/transformation.*

  - [GenApi::IFloat](#) & [Scan3dCoordinateScale](#)
- Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.*

  - [GenApi::IFloat](#) & [Scan3dCoordinateOffset](#)
- Description: Offset when transforming a pixel from relative coordinates to world coordinates.*

  - [GenApi::IBoolean](#) & [Scan3dInvalidDataFlag](#)
- Description: Enables the definition of a non-valid flag value in the data stream.*

  - [GenApi::IFloat](#) & [Scan3dInvalidDataValue](#)
- Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.*

  - [GenApi::IFloat](#) & [Scan3dAxisMin](#)
- Description: Minimum valid transmitted coordinate value of the selected Axis.*

  - [GenApi::IFloat](#) & [Scan3dAxisMax](#)
- Description: Maximum valid transmitted coordinate value of the selected Axis.*

  - [GenApi::EnumerationT< Scan3dCoordinateTransformSelectorEnums >](#) & [Scan3dCoordinateTransformSelector](#)
- Description: Sets the index to read/write a coordinate transform value.*

  - [GenApi::IFloat](#) & [Scan3dTransformValue](#)
- Description: Specifies the transform value selected.*

  - [GenApi::EnumerationT< Scan3dCoordinateReferenceSelectorEnums >](#) & [Scan3dCoordinateReferenceSelector](#)
- Description: Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.*

  - [GenApi::IFloat](#) & [Scan3dCoordinateReferenceValue](#)
- Description: Returns the reference value selected.*

  - [GenApi::Integer](#) & [ChunkPartSelector](#)
- Description: Selects the part to access in chunk data in a multipart transmission.*

  - [GenApi::EnumerationT< ChunkImageComponentEnums >](#) & [ChunkImageComponent](#)
- Description: Returns the component of the payload image.*

  - [GenApi::Integer](#) & [ChunkPixelDynamicRangeMin](#)
- Description: Returns the minimum value of dynamic range of the image included in the payload.*

  - [GenApi::Integer](#) & [ChunkPixelDynamicRangeMax](#)
- Description: Returns the maximum value of dynamic range of the image included in the payload.*

  - [GenApi::Integer](#) & [ChunkTimestampLatchValue](#)
- Description: Returns the last Timestamp latched with the TimestampLatch command.*

  - [GenApi::Integer](#) & [ChunkLineStatusAll](#)
- Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.*

  - [GenApi::EnumerationT< ChunkCounterSelectorEnums >](#) & [ChunkCounterSelector](#)
- Description: Selects which counter to retrieve data from.*



- [GenApi::Integer](#) & [ChunkCounterValue](#)  
*Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.*
- [GenApi::EnumerationT< ChunkTimerSelectorEnums >](#) & [ChunkTimerSelector](#)  
*Description: Selects which Timer to retrieve data from.*
- [GenApi::IFloat](#) & [ChunkTimerValue](#)  
*Description: Returns the value of the selected Timer at the time of the FrameStart internal event.*
- [GenApi::EnumerationT< ChunkEncoderSelectorEnums >](#) & [ChunkEncoderSelector](#)  
*Description: Selects which Encoder to retrieve data from.*
- [GenApi::Integer](#) & [ChunkScanLineSelector](#)  
*Description: Index for vector representation of one chunk value per line in an image.*
- [GenApi::Integer](#) & [ChunkEncoderValue](#)  
*Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.*
- [GenApi::EnumerationT< ChunkEncoderStatusEnums >](#) & [ChunkEncoderStatus](#)  
*Description: Returns the motion status of the selected encoder.*
- [GenApi::EnumerationT< ChunkExposureTimeSelectorEnums >](#) & [ChunkExposureTimeSelector](#)  
*Description: Selects which exposure time is read by the ChunkExposureTime feature.*
- [GenApi::Integer](#) & [ChunkLinePitch](#)  
*Description: Returns the LinePitch of the image included in the payload.*
- [GenApi::EnumerationT< ChunkSourceIDEnums >](#) & [ChunkSourceID](#)  
*Description: Returns the identifier of Source that the image comes from.*
- [GenApi::EnumerationT< ChunkRegionIDEnums >](#) & [ChunkRegionID](#)  
*Description: Returns the identifier of Region that the image comes from.*
- [GenApi::Integer](#) & [ChunkTransferBlockID](#)  
*Description: Returns the unique identifier of the transfer block used to transport the payload.*
- [GenApi::EnumerationT< ChunkTransferStreamIDEnums >](#) & [ChunkTransferStreamID](#)  
*Description: Returns identifier of the stream that generated this block.*
- [GenApi::Integer](#) & [ChunkTransferQueueCurrentBlockCount](#)  
*Description: Returns the current number of blocks in the transfer queue.*
- [GenApi::Integer](#) & [ChunkStreamChannelID](#)  
*Description: Returns identifier of the stream channel used to carry the block.*
- [GenApi::EnumerationT< ChunkScan3dDistanceUnitEnums >](#) & [ChunkScan3dDistanceUnit](#)  
*Description: Returns the Distance Unit of the payload image.*
- [GenApi::EnumerationT< ChunkScan3dOutputModeEnums >](#) & [ChunkScan3dOutputMode](#)  
*Description: Returns the Calibrated Mode of the payload image.*
- [GenApi::EnumerationT< ChunkScan3dCoordinateSystemEnums >](#) & [ChunkScan3dCoordinateSystem](#)  
*Description: Returns the Coordinate System of the image included in the payload.*
- [GenApi::EnumerationT< ChunkScan3dCoordinateSystemReferenceEnums >](#) & [ChunkScan3dCoordinateSystemReference](#)  
*Description: Returns the Coordinate System Position of the image included in the payload.*
- [GenApi::EnumerationT< ChunkScan3dCoordinateSelectorEnums >](#) & [ChunkScan3dCoordinateSelector](#)  
*Description: Selects which Coordinate to retrieve data from.*
- [GenApi::IFloat](#) & [ChunkScan3dCoordinateScale](#)  
*Description: Returns the Scale for the selected coordinate axis of the image included in the payload.*
- [GenApi::IFloat](#) & [ChunkScan3dCoordinateOffset](#)  
*Description: Returns the Offset for the selected coordinate axis of the image included in the payload.*
- [GenApi::Boolean](#) & [ChunkScan3dInvalidDataFlag](#)  
*Description: Returns if a specific non-valid data flag is used in the data stream.*
- [GenApi::IFloat](#) & [ChunkScan3dInvalidDataValue](#)  
*Description: Returns the Invalid Data Value used for the image included in the payload.*
- [GenApi::IFloat](#) & [ChunkScan3dAxisMin](#)

- Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.*

  - [GenApi::IFloat](#) & [ChunkScan3dAxisMax](#)

*Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.*

  - [GenApi::IEnumerationT](#) < [ChunkScan3dCoordinateTransformSelectorEnums](#) > & [ChunkScan3dCoordinateTransformSelector](#)

*Description: Selector for transform values.*

  - [GenApi::IFloat](#) & [ChunkScan3dTransformValue](#)

*Description: Returns the transform value.*

  - [GenApi::IEnumerationT](#) < [ChunkScan3dCoordinateReferenceSelectorEnums](#) > & [ChunkScan3dCoordinateReferenceSelector](#)

*Description: Selector to read a coordinate system reference value defining the transform of a point from one system to the other.*

  - [GenApi::IFloat](#) & [ChunkScan3dCoordinateReferenceValue](#)

*Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.*

  - [GenApi::Integer](#) & [TestPendingAck](#)

*Description: Tests the device's pending acknowledge feature.*

  - [GenApi::IEnumerationT](#) < [DeviceTapGeometryEnums](#) > & [DeviceTapGeometry](#)

*Description: This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.*

  - [GenApi::IEnumerationT](#) < [GevPhysicalLinkConfigurationEnums](#) > & [GevPhysicalLinkConfiguration](#)

*Description: Controls the principal physical link configuration to use on next restart/power-up of the device.*

  - [GenApi::IEnumerationT](#) < [GevCurrentPhysicalLinkConfigurationEnums](#) > & [GevCurrentPhysicalLinkConfiguration](#)

*Description: Indicates the current physical link configuration of the device.*

  - [GenApi::Integer](#) & [GevActiveLinkCount](#)

*Description: Indicates the current number of active logical links.*

  - [GenApi::Boolean](#) & [GevPAUSEFrameReception](#)

*Description: Controls whether incoming PAUSE Frames are handled on the given logical link.*

  - [GenApi::Boolean](#) & [GevPAUSEFrameTransmission](#)

*Description: Controls whether PAUSE Frames can be generated on the given logical link.*

  - [GenApi::IEnumerationT](#) < [GevIPConfigurationStatusEnums](#) > & [GevIPConfigurationStatus](#)

*Description: Reports the current IP configuration status.*

  - [GenApi::Integer](#) & [GevDiscoveryAckDelay](#)

*Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.*

  - [GenApi::IEnumerationT](#) < [GevGVCPExtendedStatusCodesSelectorEnums](#) > & [GevGVCPExtendedStatusCodesSelector](#)

*Description: Selects the GigE Vision version to control extended status codes for.*

  - [GenApi::Boolean](#) & [GevGVCPExtendedStatusCodes](#)

*Description: Enables the generation of extended status codes.*

  - [GenApi::Integer](#) & [GevPrimaryApplicationSwitchoverKey](#)

*Description: Controls the key to use to authenticate primary application switchover requests.*

  - [GenApi::IEnumerationT](#) < [GevGVSPExtendedIDModeEnums](#) > & [GevGVSPExtendedIDMode](#)

*Description: Enables the extended IDs mode.*

  - [GenApi::Integer](#) & [GevPrimaryApplicationSocket](#)

*Description: Returns the UDP source port of the primary application.*

  - [GenApi::Integer](#) & [GevPrimaryApplicationIPAddress](#)

*Description: Returns the address of the primary application.*

  - [GenApi::Boolean](#) & [GevSCCFGPacketResendDestination](#)

*Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.*

  - [GenApi::Boolean](#) & [GevSCCFGAllInTransmission](#)

*Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.*

- [GenApi::Integer](#) & [GevSCZoneCount](#)  
Description: Reports the number of zones per block transmitted on the selected stream channel.
- [GenApi::Integer](#) & [GevSCZoneDirectionAll](#)  
Description: Reports the transmission direction of each zone transmitted on the selected stream channel.
- [GenApi::Boolean](#) & [GevSCZoneConfigurationLock](#)  
Description: Controls whether the selected stream channel multi-zone configuration is locked.
- [GenApi::Integer](#) & [aPAUSEMACCtrlFramesTransmitted](#)  
Description: Reports the number of transmitted PAUSE frames.
- [GenApi::Integer](#) & [aPAUSEMACCtrlFramesReceived](#)  
Description: Reports the number of received PAUSE frames.
- [GenApi::EnumerationT< CIConfigurationEnums >](#) & [CIConfiguration](#)  
Description: This [Camera](#) Link specific feature describes the configuration used by the camera.
- [GenApi::EnumerationT< CITimeSlotsCountEnums >](#) & [CITimeSlotsCount](#)  
Description: This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.
- [GenApi::EnumerationT< CxpLinkConfigurationStatusEnums >](#) & [CxpLinkConfigurationStatus](#)  
Description: This feature indicates the current and active Link configuration used by the Device.
- [GenApi::EnumerationT< CxpLinkConfigurationPreferredEnums >](#) & [CxpLinkConfigurationPreferred](#)  
Description: Provides the Link configuration that allows the Transmitter Device to operate in its default mode.
- [GenApi::EnumerationT< CxpLinkConfigurationEnums >](#) & [CxpLinkConfiguration](#)  
Description: This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.
- [GenApi::Integer](#) & [CxpConnectionSelector](#)  
Description: Selects the CoaXPress physical connection to control.
- [GenApi::EnumerationT< CxpConnectionTestModeEnums >](#) & [CxpConnectionTestMode](#)  
Description: Enables the test mode for an individual physical connection of the Device.
- [GenApi::Integer](#) & [CxpConnectionTestErrorCount](#)  
Description: Reports the current connection error count for test packets recieved by the device on the connection selected by [CxpConnectionSelector](#).
- [GenApi::Integer](#) & [CxpConnectionTestPacketCount](#)  
Description: Reports the current count for test packets recieved by the device on the connection selected by [CxpConnectionSelector](#).
- [GenApi::Command](#) & [CxpPoCxpAuto](#)  
Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.
- [GenApi::Command](#) & [CxpPoCxpTurnOff](#)  
Description: Disable Power over CoaXPress (PoCXP) for the Link.
- [GenApi::Command](#) & [CxpPoCxpTripReset](#)  
Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).
- [GenApi::EnumerationT< CxpPoCxpStatusEnums >](#) & [CxpPoCxpStatus](#)  
Description: Returns the Power over CoaXPress (PoCXP) status of the Device.
- [GenApi::Integer](#) & [ChunkInferenceResult](#)  
Description: Visibility: Expert.
- [GenApi::Float](#) & [ChunkInferenceConfidence](#)  
Description: Visibility: Expert.

## Protected Member Functions

- [Camera](#) ()

## Additional Inherited Members

### 10.10.1 Detailed Description

The camera object class.

### 10.10.2 Constructor & Destructor Documentation

#### 10.10.2.1 `~Camera ( )`

#### 10.10.2.2 `Camera ( )` [protected]

### 10.10.3 Member Function Documentation

#### 10.10.3.1 `void Init ( )` [virtual]

Implements [ICameraBase](#).

### 10.10.4 Member Data Documentation

#### 10.10.4.1 `GenApi::IBoolean& AasRoiEnable`

Description:

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the AutoAlgorithm↔ Selector feature.

Visibility:

#### 10.10.4.2 `GenApi::IInteger& AasRoiHeight`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

#### 10.10.4.3 `GenApi::IInteger& AasRoiOffsetX`

Description:

Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

#### 10.10.4.4 GenApi::Integer& AasRoiOffsetY

Description:

Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

#### 10.10.4.5 GenApi::Integer& AasRoiWidth

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

#### 10.10.4.6 GenApi::ICommand& AcquisitionAbort

Description: Aborts the Acquisition immediately.

This will end the capture without completing the current Frame or waiting on a trigger. If no Acquisition is in progress, the command is ignored. Visibility: Expert

#### 10.10.4.7 GenApi::ICommand& AcquisitionArm

Description: Arms the device before an AcquisitionStart command.

This optional command validates all the current features for consistency and prepares the device for a fast start of the Acquisition. Visibility: Expert

#### 10.10.4.8 GenApi::Integer& AcquisitionBurstFrameCount

Description:

This feature is used only if the FrameBurstStart trigger is enabled and the FrameBurstEnd trigger is disabled.

Note that the total number of frames captured is also conditioned by AcquisitionFrameCount if AcquisitionMode is MultiFrame and ignored if AcquisitionMode is Single.

Visibility:

#### 10.10.4.9 **GenApi::Integer& AcquisitionFrameCount**

Description:

Number of images to acquire during a multi frame acquisition.

Visibility:

#### 10.10.4.10 **GenApi::IFloat& AcquisitionFrameRate**

Description: User controlled acquisition frame rate in Hertz Visibility:

#### 10.10.4.11 **GenApi::IBoolean& AcquisitionFrameRateEnable**

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.

Visibility:

#### 10.10.4.12 **GenApi::IFloat& AcquisitionLineRate**

Description: Controls the rate (in Hertz) at which the Lines in a Frame are captured.

Visibility:

#### 10.10.4.13 **GenApi::IEnumerationT<AcquisitionModeEnums>& AcquisitionMode**

Description: Sets the acquisition mode of the device.

Continuous: acquires images continuously. Multi Frame: acquires a specified number of images before stopping acquisition. Single Frame: acquires 1 image before stopping acquisition. Visibility:

#### 10.10.4.14 **GenApi::IFloat& AcquisitionResultingFrameRate**

Description: Resulting frame rate in Hertz.

If this does not equal the Acquisition Frame Rate it is because the Exposure Time is greater than the frame time.

Visibility:

#### 10.10.4.15 **GenApi::ICommand& AcquisitionStart**

Description: This command starts the acquisition of images.

Visibility:

**10.10.4.16 GenApi::IBoolean& AcquisitionStatus**

Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.

Visibility: Expert

**10.10.4.17 GenApi::IEnumerationT<AcquisitionStatusSelectorEnums>& AcquisitionStatusSelector**

Description: Selects the internal acquisition signal to read using AcquisitionStatus.

Visibility: Expert

**10.10.4.18 GenApi::ICommand& AcquisitionStop**

Description: This command stops the acquisition of images.

Visibility:

**10.10.4.19 GenApi::IInteger& ActionDeviceKey**

Description: Provides the device key that allows the device to check the validity of action commands.

The device internal assertion of an action signal is only authorized if the ActionDeviceKey and the action device key value in the protocol message are equal. Visibility: Guru

**10.10.4.20 GenApi::IInteger& ActionGroupKey**

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Guru

**10.10.4.21 GenApi::IInteger& ActionGroupMask**

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Guru

**10.10.4.22 GenApi::IInteger& ActionQueueSize**

Description: Indicates the size of the scheduled action commands queue.

This number represents the maximum number of scheduled action commands that can be pending at a given point in time. Visibility: Guru

#### 10.10.4.23 GenApi::Integer& ActionSelector

Description: Selects to which Action Signal further Action settings apply.

Visibility: Guru

#### 10.10.4.24 GenApi::EnumerationT<ActionUnconditionalModeEnums>& ActionUnconditionalMode

Description: Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

Visibility: Guru

#### 10.10.4.25 GenApi::Boolean& AdaptiveCompressionEnable

Description: Controls whether lossless compression adapts to the image content.

If disabled, a fixed encoding table is used. Visibility:

#### 10.10.4.26 GenApi::EnumerationT<AdcBitDepthEnums>& AdcBitDepth

Description:

Selects which ADC bit depth to use.

A higher ADC bit depth results in better image quality but slower maximum frame rate.

Visibility:

#### 10.10.4.27 GenApi::Integer& aPAUSEMACtrlFramesReceived

Description: Reports the number of received PAUSE frames.

Visibility: Guru

#### 10.10.4.28 GenApi::Integer& aPAUSEMACtrlFramesTransmitted

Description: Reports the number of transmitted PAUSE frames.

Visibility: Guru

#### 10.10.4.29 GenApi::EnumerationT<AutoAlgorithmSelectorEnums>& AutoAlgorithmSelector

Description:

Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.

Visibility:



**10.10.4.30 GenApi::IFloat& AutoExposureControlLoopDamping**

Description:

It controls how fast the exposure and gain get settled.

If the value is too small, it may cause the system to be unstable. Range is from 0.0 to 1.0. Default = 0.2.

Visibility:

**10.10.4.31 GenApi::IEnumerationT<AutoExposureControlPriorityEnums>& AutoExposureControlPriority**

Description:

Selects whether to adjust gain or exposure first.

When gain priority is selected, the camera fixes the gain to 0 dB, and the exposure is adjusted according to the target grey level. If the maximum exposure is reached before the target grey level is hit, the gain starts to change to meet the target. This mode is used to have the minimum noise. When exposure priority is selected, the camera sets the exposure to a small value (default is 5 ms). The gain is adjusted according to the target grey level. If maximum gain is reached before the target grey level is hit, the exposure starts to change to meet the target. This mode is used to capture fast motion.

Visibility:

**10.10.4.32 GenApi::IFloat& AutoExposureEVCompensation**

Description:

The EV compensation value used in the exposure compensation.

This allows you to adjust the resultant image intensity with one control. A positive value makes the image brighter. A negative value makes the image darker. Range from -3 to 3 with a step of 1/3. Default = 0.

Visibility:

**10.10.4.33 GenApi::IFloat& AutoExposureExposureTimeLowerLimit**

Description:

The smallest exposure time that auto exposure can set.

Visibility:

**10.10.4.34 GenApi::IFloat& AutoExposureExposureTimeUpperLimit**

Description:

The largest exposure time that auto exposure can set.

Visibility:

**10.10.4.35 GenApi::IFloat& AutoExposureGainLowerLimit**

Description:

The smallest gain that auto exposure can set.

Visibility:

**10.10.4.36 GenApi::IFloat& AutoExposureGainUpperLimit**

Description:

The largest gain that auto exposure can set.

Visibility:

**10.10.4.37 GenApi::IFloat& AutoExposureGreyValueLowerLimit**

Description:

The lowest value in percentage that the target mean may reach.

Visibility:

**10.10.4.38 GenApi::IFloat& AutoExposureGreyValueUpperLimit**

Description:

The highest value in percentage that the target mean may reach.

Visibility:

**10.10.4.39 GenApi::IEnumerationT<AutoExposureLightingModeEnums>& AutoExposureLightingMode**

Description:

Selects a lighting mode: Backlight, Frontlight or Normal (default).

a. Backlight compensation: used when a strong light is coming from the back of the object. b. Frontlight compensation: used when a strong light is shining in the front of the object while the background is dark. c. Normal lighting: used when the object is not under backlight or frontlight conditions. When normal lighting is selected, metering modes are available.

Visibility:

**10.10.4.40 GenApi::IEnumerationT<AutoExposureMeteringModeEnums>& AutoExposureMeteringMode**

Description:

Selects a metering mode: average, spot, or partial metering.

a. Average: Measures the light from the entire scene uniformly to determine the final exposure value. Every portion of the exposed area has the same contribution. b. Spot: Measures a small area (about 3%) in the center of the scene while the rest of the scene is ignored. This mode is used when the scene has a high contrast and the object of interest is relatively small. c. Partial: Measures the light from a larger area (about 11%) in the center of the scene. This mode is used when very dark or bright regions appear at the edge of the frame. Note: Metering mode is available only when Lighting Mode Selector is Normal.

Visibility:

**10.10.4.41 GenApi::IFloat& AutoExposureTargetGreyValue**

Description:

This is the user-specified target grey level (image mean) to apply to the current image.

Note that the target grey level is in the linear domain before gamma correction is applied.

Visibility:

**10.10.4.42 GenApi::IEnumerationT<AutoExposureTargetGreyValueAutoEnums>& AutoExposureTargetGreyValueAuto**

Description:

This indicates whether the target image grey level is automatically set by the camera or manually set by the user.

Note that the target grey level is in the linear domain before gamma correction is applied.

Visibility:

**10.10.4.43 GenApi::IFloat& BalanceRatio**

Description:

Controls the balance ratio of the selected color relative to green.

Used for white balancing.

Visibility:

**10.10.4.44 GenApi::IEnumerationT<BalanceRatioSelectorEnums>& BalanceRatioSelector**

Description:

Selects a balance ratio to configure once a balance ratio control has been selected.

Visibility:

**10.10.4.45 GenApi::IEnumerationT<BalanceWhiteAutoEnums>& BalanceWhiteAuto**

Description:

White Balance compensates for color shifts caused by different lighting conditions.

It can be automatically or manually controlled. For manual control, set to Off. For automatic control, set to Once or Continuous.

Visibility:

**10.10.4.46 GenApi::IFloat& BalanceWhiteAutoDamping**

Description:

Controls how quickly 'BalanceWhiteAuto' adjusts the values for Red and Blue BalanceRatio in response to changing conditions.

Higher damping means the changes are more gradual.

Visibility:

**10.10.4.47 GenApi::IFloat& BalanceWhiteAutoLowerLimit**

Description:

Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.

Visibility:

**10.10.4.48 GenApi::IEnumerationT<BalanceWhiteAutoProfileEnums>& BalanceWhiteAutoProfile**

Description: Selects the profile used by BalanceWhiteAuto.

Visibility:

**10.10.4.49 GenApi::IFloat& BalanceWhiteAutoUpperLimit**

Description:

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

Visibility:

**10.10.4.50 GenApi::Integer& BinningHorizontal**

Description:

Number of horizontal photo-sensitive cells to combine together.

This reduces the horizontal resolution (width) of the image. A value of 1 indicates that no horizontal binning is performed by the camera. This value must be 1 for decimation to be active.

Visibility:

**10.10.4.51 GenApi::IEnumerationT<BinningHorizontalModeEnums>& BinningHorizontalMode**

Description: Visibility:

**10.10.4.52 GenApi::IEnumerationT<BinningSelectorEnums>& BinningSelector**

Description:

Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.

Visibility:

**10.10.4.53 GenApi::Integer& BinningVertical**

Description:

Number of vertical photo-sensitive cells to combine together.

This reduces the vertical resolution (height) of the image. A value of 1 indicates that no vertical binning is performed by the camera. This value must be 1 for decimation to be active.

Visibility:

**10.10.4.54 GenApi::IEnumerationT<BinningVerticalModeEnums>& BinningVerticalMode**

Description: Visibility:

#### 10.10.4.55 GenApi::IFloat& BlackLevel

Description:

Controls the offset of the video signal in percent.

Visibility:

#### 10.10.4.56 GenApi::IEnumerationT<BlackLevelAutoEnums>& BlackLevelAuto

Description: Controls the mode for automatic black level adjustment.

The exact algorithm used to implement this adjustment is device-specific. Visibility: Expert

#### 10.10.4.57 GenApi::IEnumerationT<BlackLevelAutoBalanceEnums>& BlackLevelAutoBalance

Description: Controls the mode for automatic black level balancing between the sensor color channels or taps.

The black level coefficients of each channel are adjusted so they are matched. Visibility: Expert

#### 10.10.4.58 GenApi::IBoolean& BlackLevelClampingEnable

Description:

Enable the black level auto clamping feature which performing dark current compensation.

Visibility:

#### 10.10.4.59 GenApi::IInteger& BlackLevelRaw

Description:

Controls the offset of the video signal in camera specific units.

Visibility:

#### 10.10.4.60 GenApi::IEnumerationT<BlackLevelSelectorEnums>& BlackLevelSelector

Description:

Selects which black level to control.

Only All can be set by the user. Analog and Digital are read-only.

Visibility:

**10.10.4.61 GenApi::IFloat& ChunkBlackLevel**

Description: Returns the black level used to capture the image.

Visibility:

**10.10.4.62 GenApi::IEnumerationT<ChunkBlackLevelSelectorEnums>& ChunkBlackLevelSelector**

Description: Selects which black level to retrieve Visibility:

**10.10.4.63 GenApi::IEnumerationT<ChunkCounterSelectorEnums>& ChunkCounterSelector**

Description: Selects which counter to retrieve data from.

Visibility: Expert

**10.10.4.64 GenApi::IInteger& ChunkCounterValue**

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

**10.10.4.65 GenApi::IInteger& ChunkCRC**

Description: Returns the CRC of the image payload.

Visibility:

**10.10.4.66 GenApi::IBoolean& ChunkEnable**

Description: Enables the inclusion of the selected Chunk data in the payload of the image.

Visibility:

**10.10.4.67 GenApi::IEnumerationT<ChunkEncoderSelectorEnums>& ChunkEncoderSelector**

Description: Selects which Encoder to retrieve data from.

Visibility: Expert

**10.10.4.68 GenApi::IEnumerationT<ChunkEncoderStatusEnums>& ChunkEncoderStatus**

Description: Returns the motion status of the selected encoder.

Visibility: Expert

**10.10.4.69 GenApi::Integer& ChunkEncoderValue**

Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.

Visibility: Expert

**10.10.4.70 GenApi::Integer& ChunkExposureEndLineStatusAll**

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

**10.10.4.71 GenApi::Float& ChunkExposureTime**

Description: Returns the exposure time used to capture the image.

Visibility:

**10.10.4.72 GenApi::EnumerationT<ChunkExposureTimeSelectorEnums>& ChunkExposureTimeSelector**

Description: Selects which exposure time is read by the ChunkExposureTime feature.

Visibility: Expert

**10.10.4.73 GenApi::Integer& ChunkFrameID**

Description: Returns the image frame ID.

Visibility:

**10.10.4.74 GenApi::Float& ChunkGain**

Description: Returns the gain used to capture the image.

Visibility:

**10.10.4.75 GenApi::EnumerationT<ChunkGainSelectorEnums>& ChunkGainSelector**

Description: Selects which gain to retrieve

**10.10.4.76 GenApi::Integer& ChunkHeight**

Description: Returns the height of the image included in the payload.

Visibility:



**10.10.4.77 GenApi::Integer& ChunkImage**

Description: Returns the image payload.

Visibility:

**10.10.4.78 GenApi::EnumerationT<ChunkImageComponentEnums>& ChunkImageComponent**

Description: Returns the component of the payload image.

This can be used to identify the image component of a generic part in a multipart transfer. Visibility: Expert

**10.10.4.79 GenApi::Float& ChunkInferenceConfidence**

Description: Visibility: Expert.

**10.10.4.80 GenApi::Integer& ChunkInferenceResult**

Description: Visibility: Expert.

**10.10.4.81 GenApi::Integer& ChunkLinePitch**

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

**10.10.4.82 GenApi::Integer& ChunkLineStatusAll**

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

**10.10.4.83 GenApi::Boolean& ChunkModeActive**

Description: Activates the inclusion of Chunk data in the payload of the image.

Visibility:

**10.10.4.84 GenApi::Integer& ChunkOffsetX**

Description: Returns the Offset X of the image included in the payload.

Visibility:

**10.10.4.85 GenApi::Integer& ChunkOffsetY**

Description: Returns the Offset Y of the image included in the payload.

Visibility:

**10.10.4.86 GenApi::Integer& ChunkPartSelector**

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

**10.10.4.87 GenApi::Integer& ChunkPixelDynamicRangeMax**

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

**10.10.4.88 GenApi::Integer& ChunkPixelDynamicRangeMin**

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

**10.10.4.89 GenApi::EnumerationT<ChunkPixelFormatEnums>& ChunkPixelFormat**

Description: Format of the pixel provided by the camera Visibility:

**10.10.4.90 GenApi::EnumerationT<ChunkRegionIDEnums>& ChunkRegionID**

Description: Returns the identifier of Region that the image comes from.

Visibility: Expert

**10.10.4.91 GenApi::IFloat& ChunkScan3dAxisMax**

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

**10.10.4.92 GenApi::IFloat& ChunkScan3dAxisMin**

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

**10.10.4.93 GenApi::IFloat& ChunkScan3dCoordinateOffset**

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

**10.10.4.94 GenApi::IEnumerationT<ChunkScan3dCoordinateReferenceSelectorEnums>& ChunkScan3dCoordinateReferenceSelector**

Description: Selector to read a coordinate system reference value defining the transform of a point from one system to the other.

Visibility: Expert

**10.10.4.95 GenApi::IFloat& ChunkScan3dCoordinateReferenceValue**

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.

Visibility: Expert

**10.10.4.96 GenApi::IFloat& ChunkScan3dCoordinateScale**

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

**10.10.4.97 GenApi::IEnumerationT<ChunkScan3dCoordinateSelectorEnums>& ChunkScan3dCoordinateSelector**

Description: Selects which Coordinate to retrieve data from.

Visibility: Expert

**10.10.4.98 GenApi::IEnumerationT<ChunkScan3dCoordinateSystemEnums>& ChunkScan3dCoordinateSystem**

Description: Returns the Coordinate [System](#) of the image included in the payload.

Visibility: Expert

**10.10.4.99 GenApi::IEnumerationT<ChunkScan3dCoordinateSystemReferenceEnums>& ChunkScan3dCoordinateSystemReference**

Description: Returns the Coordinate [System](#) Position of the image included in the payload.

Visibility: Expert

#### 10.10.4.100 **GenApi::IEnumerationT<ChunkScan3dCoordinateTransformSelectorEnums>& ChunkScan3dCoordinateTransformSelector**

Description: Selector for transform values.

Visibility: Expert

#### 10.10.4.101 **GenApi::IEnumerationT<ChunkScan3dDistanceUnitEnums>& ChunkScan3dDistanceUnit**

Description: Returns the Distance Unit of the payload image.

Visibility: Expert

#### 10.10.4.102 **GenApi::IBoolean& ChunkScan3dInvalidDataFlag**

Description: Returns if a specific non-valid data flag is used in the data stream.

Visibility: Expert

#### 10.10.4.103 **GenApi::IFloat& ChunkScan3dInvalidDataValue**

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

#### 10.10.4.104 **GenApi::IEnumerationT<ChunkScan3dOutputModeEnums>& ChunkScan3dOutputMode**

Description: Returns the Calibrated Mode of the payload image.

Visibility: Expert

#### 10.10.4.105 **GenApi::IFloat& ChunkScan3dTransformValue**

Description: Returns the transform value.

Visibility: Expert

#### 10.10.4.106 **GenApi::IInteger& ChunkScanLineSelector**

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

**10.10.4.107 GenApi::IEnumerationT<ChunkSelectorEnums>& ChunkSelector**

Description: Selects which chunk data to enable or disable.

Visibility:

**10.10.4.108 GenApi::Integer& ChunkSequencerSetActive**

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

**10.10.4.109 GenApi::IString& ChunkSerialData**

Description: Returns the serial data that was received.

Visibility:

**10.10.4.110 GenApi::Integer& ChunkSerialDataLength**

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

**10.10.4.111 GenApi::IBoolean& ChunkSerialReceiveOverflow**

Description: Returns the status of the chunk serial receive overflow.

Visibility:

**10.10.4.112 GenApi::IEnumerationT<ChunkSourceIDEnums>& ChunkSourceID**

Description: Returns the identifier of Source that the image comes from.

Visibility: Expert

**10.10.4.113 GenApi::Integer& ChunkStreamChannelID**

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

**10.10.4.114 GenApi::IEnumerationT<ChunkTimerSelectorEnums>& ChunkTimerSelector**

Description: Selects which Timer to retrieve data from.

Visibility: Expert

**10.10.4.115 GenApi::IFloat& ChunkTimerValue**

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

**10.10.4.116 GenApi::Integer& ChunkTimestamp**

Description: Returns the Timestamp of the image.

Visibility:

**10.10.4.117 GenApi::Integer& ChunkTimestampLatchValue**

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

**10.10.4.118 GenApi::Integer& ChunkTransferBlockID**

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

**10.10.4.119 GenApi::Integer& ChunkTransferQueueCurrentBlockCount**

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

**10.10.4.120 GenApi::EnumerationT<ChunkTransferStreamIDEnums>& ChunkTransferStreamID**

Description: Returns identifier of the stream that generated this block.

Visibility: Expert

**10.10.4.121 GenApi::Integer& ChunkWidth**

Description: Returns the width of the image included in the payload.

Visibility:

**10.10.4.122 GenApi::IEnumerationT<CIConfigurationEnums>& CIConfiguration**

Description: This [Camera](#) Link specific feature describes the configuration used by the camera.

It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitizationTaps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera. Visibility: Beginner

**10.10.4.123 GenApi::IEnumerationT<CITimeSlotsCountEnums>& CITimeSlotsCount**

Description: This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

Visibility: Expert

**10.10.4.124 GenApi::IBoolean& ColorTransformationEnable**

Description:

Enables/disables the color transform selected with ColorTransformationSelector.

For RGB to YUV this is read-only. Enabling/disabling RGB to YUV can only be done by changing pixel format.

Visibility:

**10.10.4.125 GenApi::IEnumerationT<ColorTransformationSelectorEnums>& ColorTransformationSelector**

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.

Visibility:

**10.10.4.126 GenApi::IFloat& ColorTransformationValue**

Description:

Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.

Visibility:

**10.10.4.127 GenApi::IEnumerationT<ColorTransformationValueSelectorEnums>& ColorTransformationValueSelector**

Description:

Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module

Visibility:

**10.10.4.128 GenApi::IFloat& CompressionRatio**

Description: Reports the ratio between the uncompressed image size and compressed image size.

Visibility:

**10.10.4.129 GenApi::Integer& CounterDelay**

Description: Sets the delay (or number of events) before the CounterStart event is generated.

Visibility:

**10.10.4.130 GenApi::Integer& CounterDuration**

Description: Sets the duration (or number of events) before the CounterEnd event is generated.

Visibility:

**10.10.4.131 GenApi::EnumerationT<CounterEventActivationEnums>& CounterEventActivation**

Description: Selects the activation mode of the event to increment the Counter.

Visibility:

**10.10.4.132 GenApi::EnumerationT<CounterEventSourceEnums>& CounterEventSource**

Description: Selects the event that will increment the counter Visibility:

**10.10.4.133 GenApi::ICommand& CounterReset**

Description: Does a software reset of the selected Counter and starts it.

The counter starts counting events immediately after the reset unless a Counter trigger is active. CounterReset can be used to reset the Counter independently from the CounterResetSource. To disable the counter temporarily, set CounterEventSource to Off. Visibility: Expert

**10.10.4.134 GenApi::EnumerationT<CounterResetActivationEnums>& CounterResetActivation**

Description: Selects the Activation mode of the Counter Reset Source signal.

Visibility:

**10.10.4.135 GenApi::EnumerationT<CounterResetSourceEnums>& CounterResetSource**

Description: Selects the signal that will be the source to reset the Counter.

Visibility:



**10.10.4.136 GenApi::IEnumerationT<CounterSelectorEnums> & CounterSelector**

Description: Selects which counter to configure Visibility:

**10.10.4.137 GenApi::IEnumerationT<CounterStatusEnums> & CounterStatus**

Description: Returns the current status of the Counter.

Visibility:

**10.10.4.138 GenApi::IEnumerationT<CounterTriggerActivationEnums> & CounterTriggerActivation**

Description: Selects the activation mode of the trigger to start the Counter.

Visibility:

**10.10.4.139 GenApi::IEnumerationT<CounterTriggerSourceEnums> & CounterTriggerSource**

Description: Selects the source of the trigger to start the counter Visibility:

**10.10.4.140 GenApi::Integer & CounterValue**

Description: Current counter value Visibility:

**10.10.4.141 GenApi::Integer & CounterValueAtReset**

Description: Value of the selected Counter when it was reset by a trigger.

Visibility:

**10.10.4.142 GenApi::Integer & CxpConnectionSelector**

Description: Selects the CoaXPress physical connection to control.

Visibility: Expert

**10.10.4.143 GenApi::Integer & CxpConnectionTestErrorCount**

Description: Reports the current connection error count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

**10.10.4.144 GenApi::IEnumerationT<CxpConnectionTestModeEnums>& CxpConnectionTestMode**

Description: Enables the test mode for an individual physical connection of the Device.

Visibility: Expert

**10.10.4.145 GenApi::Integer& CxpConnectionTestPacketCount**

Description: Reports the current count for test packets recieved by the device on the connection selected by Cxp↔ConnectionSelector.

Visibility: Expert

**10.10.4.146 GenApi::IEnumerationT<CxpLinkConfigurationEnums>& CxpLinkConfiguration**

Description: This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.

In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus. Visibility: Beginner

**10.10.4.147 GenApi::IEnumerationT<CxpLinkConfigurationPreferredEnums>& CxpLinkConfigurationPreferred**

Description: Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

Visibility: Expert

**10.10.4.148 GenApi::IEnumerationT<CxpLinkConfigurationStatusEnums>& CxpLinkConfigurationStatus**

Description: This feature indicates the current and active Link configuration used by the Device.

Visibility: Beginner

**10.10.4.149 GenApi::ICommand& CxpPoCxpAuto**

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

**10.10.4.150 GenApi::IEnumerationT<CxpPoCxpStatusEnums>& CxpPoCxpStatus**

Description: Returns the Power over CoaXPress (PoCXP) status of the Device.

Visibility: Expert

**10.10.4.151 GenApi::ICommand& CxpPoCxpTripReset**

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

Visibility: Expert

**10.10.4.152 GenApi::ICommand& CxpPoCxpTurnOff**

Description: Disable Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

**10.10.4.153 GenApi::Integer& DecimationHorizontal**

Description:

Horizontal decimation of the image.

This reduces the horizontal resolution (width) of the image by only retaining a single pixel within a window whose size is the decimation factor specified here. A value of 1 indicates that no horizontal decimation is performed by the camera. This value must be 1 for binning to be active.

Visibility:

**10.10.4.154 GenApi::IEnumerationT<DecimationHorizontalModeEnums>& DecimationHorizontalMode**

Description:

The mode used to reduce the horizontal resolution when DecimationHorizontal is used.

The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Visibility:

**10.10.4.155 GenApi::IEnumerationT<DecimationSelectorEnums>& DecimationSelector**

Description: Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.

Visibility:

#### 10.10.4.156 GenApi::Integer& DecimationVertical

Description:

Vertical decimation of the image.

This reduces the vertical resolution (height) of the image by only retaining a single pixel within a window whose size is the decimation factor specified here. A value of 1 indicates that no vertical decimation is performed by the camera. This value must be 1 for binning to be active.

Visibility:

#### 10.10.4.157 GenApi::IEnumerationT<DecimationVerticalModeEnums>& DecimationVerticalMode

Description:

The mode used to reduce the vertical resolution when DecimationVertical is used.

The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Visibility:

#### 10.10.4.158 GenApi::IEnumerationT<DefectCorrectionModeEnums>& DefectCorrectionMode

Description: Controls the method used for replacing defective pixels.

Visibility:

#### 10.10.4.159 GenApi::IBoolean& DefectCorrectStaticEnable

Description: Enables/Disables table-based defective pixel correction.

Visibility:

#### 10.10.4.160 GenApi::ICommand& DefectTableApply

Description: Applies the current defect table, so that any changes made affect images captured by the camera.

This writes the table to volatile memory, so changes to the table are lost if the camera loses power. To save the table to non-volatile memory, use DefectTableSave.

Visibility:

**10.10.4.161 GenApi::Integer& DefectTableCoordinateX**

Description:

Returns the X coordinate of the defective pixel at DefectTableIndex within the defective pixel table.

Changes made do not take effect in captured images until the command DefectTableApply is written.

Visibility:

**10.10.4.162 GenApi::Integer& DefectTableCoordinateY**

Description:

Returns the Y coordinate of the defective pixel at DefectTableIndex within the defective pixel table.

Changes made do not take effect in captured images until the command DefectTableApply is written.

Visibility:

**10.10.4.163 GenApi::ICommand& DefectTableFactoryRestore**

Description: Restores the Defective Pixel Table to its factory default state, which was calibrated during manufacturing.

This permanently overwrites any changes made to the defect table.

Visibility:

**10.10.4.164 GenApi::Integer& DefectTableIndex**

Description:

Controls the offset of the element to access in the defective pixel location table.

Visibility:

**10.10.4.165 GenApi::Integer& DefectTablePixelCount**

Description:

The number of defective pixel locations in the current table.

Visibility:

**10.10.4.166 GenApi::ICommand& DefectTableSave**

Description: Saves the current defective pixel table non-volatile memory, so that it is preserved when the camera boots up.

This overwrites the existing defective pixel table. The new table is loaded whenever the camera powers up.

Visibility:

**10.10.4.167 GenApi::IEnumerationT<DeinterlacingEnums>& Deinterlacing**

Description: Controls how the device performs de-interlacing.

Visibility: Beginner

**10.10.4.168 GenApi::IEnumerationT<DeviceCharacterSetEnums>& DeviceCharacterSet**

Description:

Character set used by the strings of the device's bootstrap registers.

Visibility:

**10.10.4.169 GenApi::IFloat& DeviceClockFrequency**

Description: Returns the frequency of the selected Clock.

Visibility: Expert

**10.10.4.170 GenApi::IEnumerationT<DeviceClockSelectorEnums>& DeviceClockSelector**

Description: Selects the clock frequency to access from the device.

Visibility: Expert

**10.10.4.171 GenApi::Integer& DeviceConnectionSelector**

Description: Selects which Connection of the device to control.

Visibility: Beginner

**10.10.4.172 GenApi::Integer& DeviceConnectionSpeed**

Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.

**10.10.4.173 GenApi::IEnumerationT<DeviceConnectionStatusEnums>& DeviceConnectionStatus**

Description: Indicates the status of the specified Connection.

Visibility: Expert

**10.10.4.174 GenApi::Integer& DeviceEventChannelCount**

Description:

Indicates the number of event channels supported by the device.

Visibility:

**10.10.4.175 GenApi::IString& DeviceFamilyName**

Description: Identifier of the product family of the device.

Visibility: Beginner

**10.10.4.176 GenApi::ICommand& DeviceFeaturePersistenceEnd**

Description: Indicate to the device the end of feature persistence.

Visibility: Guru

**10.10.4.177 GenApi::ICommand& DeviceFeaturePersistenceStart**

Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.

Visibility: Guru

**10.10.4.178 GenApi::IString& DeviceFirmwareVersion**

Description: Version of the firmware on the device.

Visibility:

**10.10.4.179 GenApi::Integer& DeviceGenCPVersionMajor**

Description: Major version of the GenCP protocol supported by the device.

Visibility: Beginner

**10.10.4.180 GenApi::Integer& DeviceGenCPVersionMinor**

Description: Minor version of the GenCP protocol supported by the device.

Visibility: Beginner

**10.10.4.181 GenApi::IString& DeviceID**

Description: Device identifier (serial number).

Visibility:

**10.10.4.182 GenApi::IEnumerationT<DeviceIndicatorModeEnums>& DeviceIndicatorMode**

Description: Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).

Visibility:

**10.10.4.183 GenApi::IFloat& DeviceLinkBandwidthReserve**

Description:

Percentage of streamed data bandwidth reserved for packet resend.

Visibility:

**10.10.4.184 GenApi::IFloat& DeviceLinkCommandTimeout**

Description: Indicates the command timeout of the specified Link.

This corresponds to the maximum response time of the device for a command sent on that link. Visibility: Guru

**10.10.4.185 GenApi::Integer& DeviceLinkConnectionCount**

Description: Returns the number of physical connection of the device used by a particular Link.

Visibility: Beginner

**10.10.4.186 GenApi::Integer& DeviceLinkCurrentThroughput**

Description: Current bandwidth of streamed data.

Visibility:



**10.10.4.187 GenApi::IEnumerationT<DeviceLinkHeartbeatModeEnums>& DeviceLinkHeartbeatMode**

Description: Activate or deactivate the Link's heartbeat.

Visibility: Expert

**10.10.4.188 GenApi::IFloat& DeviceLinkHeartbeatTimeout**

Description: Controls the current heartbeat timeout of the specific Link.

Visibility: Guru

**10.10.4.189 GenApi::IInteger& DeviceLinkSelector**

Description: Selects which Link of the device to control.

Visibility: Beginner

**10.10.4.190 GenApi::IInteger& DeviceLinkSpeed**

Description:

Indicates the speed of transmission negotiated on the specified Link.

(Bps)

Visibility:

**10.10.4.191 GenApi::IInteger& DeviceLinkThroughputLimit**

Description:

Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link.

If necessary, delays will be uniformly inserted between transport layer packets in order to control the peak bandwidth.

Visibility:

**10.10.4.192 GenApi::IEnumerationT<DeviceLinkThroughputLimitModeEnums>& DeviceLinkThroughputLimitMode**

Description: Controls if the DeviceLinkThroughputLimit is active.

When disabled, lower level TL specific features are expected to control the throughput. When enabled, Device↔LinkThroughputLimit controls the overall throughput. Visibility: Expert

**10.10.4.193 GenApi::Integer& DeviceManifestEntrySelector**

Description: Selects the manifest entry to reference.

Visibility: Guru

**10.10.4.194 GenApi::IString& DeviceManifestPrimaryURL**

Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

**10.10.4.195 GenApi::Integer& DeviceManifestSchemaMajorVersion**

Description: Indicates the major version number of the schema file of the selected manifest entry.

Visibility: Guru

**10.10.4.196 GenApi::Integer& DeviceManifestSchemaMinorVersion**

Description: Indicates the minor version number of the schema file of the selected manifest entry.

Visibility: Guru

**10.10.4.197 GenApi::IString& DeviceManifestSecondaryURL**

Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

**10.10.4.198 GenApi::Integer& DeviceManifestXMLMajorVersion**

Description: Indicates the major version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

**10.10.4.199 GenApi::Integer& DeviceManifestXMLMinorVersion**

Description: Indicates the minor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

**10.10.4.200 GenApi::Integer& DeviceManifestXMLSubMinorVersion**

Description: Indicates the subminor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

**10.10.4.201 GenApi::IString& DeviceManufacturerInfo**

Description: Manufacturer information about the device.

Visibility:

**10.10.4.202 GenApi::Integer& DeviceMaxThroughput**

Description:

Maximum bandwidth of the data that can be streamed out of the device.

This can be used to estimate if the physical connection(s) can sustain transfer of free-running images from the camera at its maximum speed.

Visibility:

**10.10.4.203 GenApi::IString& DeviceModelName**

Description: Model of the device.

Visibility:

**10.10.4.204 GenApi::IEnumerationT<DevicePowerSupplySelectorEnums>& DevicePowerSupplySelector**

Description:

Selects the power supply source to control or read.

Visibility:

**10.10.4.205 GenApi::ICommand& DeviceRegistersCheck**

Description: Perform the validation of the current register set for consistency.

This will update the DeviceRegistersValid flag. Visibility: Expert

**10.10.4.206 GenApi::IEnumerationT<DeviceRegistersEndiannessEnums>& DeviceRegistersEndianness**

Description: Endianness of the registers of the device.

Visibility:

**10.10.4.207 GenApi::ICommand& DeviceRegistersStreamingEnd**

Description: Announce the end of registers streaming.

This will do a register set validation for consistency and activate it. This will also update the DeviceRegistersValid flag. Visibility: Guru

**10.10.4.208 GenApi::ICommand& DeviceRegistersStreamingStart**

Description: Prepare the device for registers streaming without checking for consistency.

Visibility: Guru

**10.10.4.209 GenApi::IBoolean& DeviceRegistersValid**

Description: Returns if the current register set is valid and consistent.

Visibility: Expert

**10.10.4.210 GenApi::ICommand& DeviceReset**

Description: This is a command that immediately resets and reboots the device.

Visibility:

**10.10.4.211 GenApi::IEnumerationT<DeviceScanTypeEnums>& DeviceScanType**

Description: Scan type of the sensor of the device.

Visibility:

**10.10.4.212 GenApi::IString& DeviceSerialNumber**

Description:

Device's serial number.

This string is a unique identifier of the device.

Visibility:

**10.10.4.213 GenApi::IEnumerationT<DeviceSerialPortBaudRateEnums>& DeviceSerialPortBaudRate**

Description: This feature controls the baud rate used by the selected serial port.

Visibility: Expert

**10.10.4.214 GenApi::IEnumerationT<DeviceSerialPortSelectorEnums> & DeviceSerialPortSelector**

Description: Selects which serial port of the device to control.

Visibility: Expert

**10.10.4.215 GenApi::Integer & DeviceSFNCVersionMajor**

Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

**10.10.4.216 GenApi::Integer & DeviceSFNCVersionMinor**

Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

**10.10.4.217 GenApi::Integer & DeviceSFNCVersionSubMinor**

Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

**10.10.4.218 GenApi::Integer & DeviceStreamChannelCount**

Description:

Indicates the number of streaming channels supported by the device.

Visibility:

**10.10.4.219 GenApi::IEnumerationT<DeviceStreamChannelEndiannessEnums> & DeviceStreamChannelEndianness**

Description: Endianness of multi-byte pixel data for this stream.

Visibility: Guru

**10.10.4.220 GenApi::Integer & DeviceStreamChannelLink**

Description: Index of device's Link to use for streaming the specified stream channel.

Visibility: Guru

**10.10.4.221 GenApi::Integer& DeviceStreamChannelPacketSize**

Description: Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.

Visibility: Expert

**10.10.4.222 GenApi::Integer& DeviceStreamChannelSelector**

Description: Selects the stream channel to control.

Visibility: Expert

**10.10.4.223 GenApi::EnumerationT<DeviceStreamChannelTypeEnums>& DeviceStreamChannelType**

Description: Reports the type of the stream channel.

Visibility: Guru

**10.10.4.224 GenApi::EnumerationT<DeviceTapGeometryEnums>& DeviceTapGeometry**

Description: This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

Visibility: Expert

**10.10.4.225 GenApi::IFloat& DeviceTemperature**

Description: Device temperature in degrees Celsius (C).

Visibility:

**10.10.4.226 GenApi::EnumerationT<DeviceTemperatureSelectorEnums>& DeviceTemperatureSelector**

Description:

Selects the location within the device, where the temperature will be measured.

Visibility:

**10.10.4.227 GenApi::EnumerationT<DeviceTLTypeEnums>& DeviceTLType**

Description: Transport Layer type of the device.

Visibility:

**10.10.4.228 GenApi::Integer& DeviceTLVersionMajor**

Description:

Major version of the Transport Layer of the device.

Visibility:

**10.10.4.229 GenApi::Integer& DeviceTLVersionMinor**

Description:

Minor version of the Transport Layer of the device.

Visibility:

**10.10.4.230 GenApi::Integer& DeviceTLVersionSubMinor**

Description: Sub minor version of the Transport Layer of the device.

Visibility: Beginner

**10.10.4.231 GenApi::EnumerationT<DeviceTypeEnums>& DeviceType**

Description: Returns the device type.

Visibility: Guru

**10.10.4.232 GenApi::Integer& DeviceUptime**

Description: Total time since the device was powered up in seconds.

Visibility:

**10.10.4.233 GenApi::IString& DeviceUserID**

Description: User-programmable device identifier.

Visibility:

**10.10.4.234 GenApi::IString& DeviceVendorName**

Description: Name of the manufacturer of the device.

Visibility:

**10.10.4.235 GenApi::IString& DeviceVersion**

Description: Version of the device.

Visibility:

**10.10.4.236 GenApi::IInteger& EncoderDivider**

Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.

Visibility: Expert

**10.10.4.237 GenApi::IEnumerationT<EncoderModeEnums>& EncoderMode**

Description: Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.

Visibility: Expert

**10.10.4.238 GenApi::IEnumerationT<EncoderOutputModeEnums>& EncoderOutputMode**

Description: Selects the conditions for the Encoder interface to generate a valid Encoder output signal.

Visibility: Expert

**10.10.4.239 GenApi::ICommand& EncoderReset**

Description: Does a software reset of the selected Encoder and starts it.

The Encoder starts counting events immediately after the reset. EncoderReset can be used to reset the Encoder independently from the EncoderResetSource. Visibility: Expert

**10.10.4.240 GenApi::IEnumerationT<EncoderResetActivationEnums>& EncoderResetActivation**

Description: Selects the Activation mode of the Encoder Reset Source signal.

Visibility: Expert

**10.10.4.241 GenApi::IEnumerationT<EncoderResetSourceEnums>& EncoderResetSource**

Description: Selects the signals that will be the source to reset the Encoder.

Visibility: Expert



**10.10.4.242 GenApi::IEnumerationT<EncoderSelectorEnums>& EncoderSelector**

Description: Selects which Encoder to configure.

Visibility: Expert

**10.10.4.243 GenApi::IEnumerationT<EncoderSourceAEnums>& EncoderSourceA**

Description: Selects the signal which will be the source of the A input of the Encoder.

Visibility: Expert

**10.10.4.244 GenApi::IEnumerationT<EncoderSourceBEnums>& EncoderSourceB**

Description: Selects the signal which will be the source of the B input of the Encoder.

Visibility: Expert

**10.10.4.245 GenApi::IEnumerationT<EncoderStatusEnums>& EncoderStatus**

Description: Returns the motion status of the encoder.

Visibility: Expert

**10.10.4.246 GenApi::IFloat& EncoderTimeout**

Description: Sets the maximum time interval between encoder counter increments before the status turns to static.

Visibility: Expert

**10.10.4.247 GenApi::Integer& EncoderValue**

Description: Reads or writes the current value of the position counter of the selected Encoder.

Visibility: Expert

**10.10.4.248 GenApi::Integer& EncoderValueAtReset**

Description: Reads the value of the of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.

Visibility: Expert

**10.10.4.249 GenApi::Integer& EnumerationCount**

Description: Number of enumerations since uptime.

Visibility:

**10.10.4.250 GenApi::Integer& EventAcquisitionEnd**

Description: Returns the unique Identifier of the Acquisition End type of [Event](#).

Visibility: Expert

**10.10.4.251 GenApi::Integer& EventAcquisitionEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End [Event](#).

Visibility: Expert

**10.10.4.252 GenApi::Integer& EventAcquisitionEndTimestamp**

Description: Returns the Timestamp of the Acquisition End [Event](#).

Visibility: Expert

**10.10.4.253 GenApi::Integer& EventAcquisitionError**

Description: Returns the unique Identifier of the Acquisition Error type of [Event](#).

Visibility: Expert

**10.10.4.254 GenApi::Integer& EventAcquisitionErrorFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error [Event](#).

Visibility: Expert

**10.10.4.255 GenApi::Integer& EventAcquisitionErrorTimestamp**

Description: Returns the Timestamp of the Acquisition Error [Event](#).

Visibility: Expert

**10.10.4.256 GenApi::Integer& EventAcquisitionStart**

Description: Returns the unique Identifier of the Acquisition Start type of [Event](#).

Visibility: Expert

**10.10.4.257 GenApi::Integer& EventAcquisitionStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start [Event](#).

Visibility: Expert

**10.10.4.258 GenApi::Integer& EventAcquisitionStartTimestamp**

Description: Returns the Timestamp of the Acquisition Start [Event](#).

Visibility: Expert

**10.10.4.259 GenApi::Integer& EventAcquisitionTransferEnd**

Description: Returns the unique Identifier of the Acquisition Transfer End type of [Event](#).

Visibility: Expert

**10.10.4.260 GenApi::Integer& EventAcquisitionTransferEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End [Event](#).

Visibility: Expert

**10.10.4.261 GenApi::Integer& EventAcquisitionTransferEndTimestamp**

Description: Returns the Timestamp of the Acquisition Transfer End [Event](#).

Visibility: Expert

**10.10.4.262 GenApi::Integer& EventAcquisitionTransferStart**

Description: Returns the unique Identifier of the Acquisition Transfer Start type of [Event](#).

Visibility: Expert

**10.10.4.263 GenApi::Integer& EventAcquisitionTransferStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start [Event](#).

Visibility: Expert

**10.10.4.264 GenApi::Integer& EventAcquisitionTransferStartTimestamp**

Description: Returns the Timestamp of the Acquisition Transfer Start [Event](#).

Visibility: Expert

**10.10.4.265 GenApi::Integer& EventAcquisitionTrigger**

Description: Returns the unique Identifier of the Acquisition Trigger type of [Event](#).

Visibility: Expert

**10.10.4.266 GenApi::Integer& EventAcquisitionTriggerFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger [Event](#).

Visibility: Expert

**10.10.4.267 GenApi::Integer& EventAcquisitionTriggerTimestamp**

Description: Returns the Timestamp of the Acquisition Trigger [Event](#).

Visibility: Expert

**10.10.4.268 GenApi::Integer& EventActionLate**

Description: Returns the unique Identifier of the Action Late type of [Event](#).

Visibility: Expert

**10.10.4.269 GenApi::Integer& EventActionLateFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late [Event](#).

Visibility: Expert

**10.10.4.270 GenApi::Integer& EventActionLateTimestamp**

Description: Returns the Timestamp of the Action Late [Event](#).

Visibility: Expert

**10.10.4.271 GenApi::Integer& EventCounter0End**

Description: Returns the unique Identifier of the Counter 0 End type of [Event](#).

Visibility: Expert

**10.10.4.272 GenApi::Integer& EventCounter0EndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End [Event](#).

Visibility: Expert

**10.10.4.273 GenApi::Integer& EventCounter0EndTimestamp**

Description: Returns the Timestamp of the Counter 0 End [Event](#).

Visibility: Expert

**10.10.4.274 GenApi::Integer& EventCounter0Start**

Description: Returns the unique Identifier of the Counter 0 Start type of [Event](#).

Visibility: Expert

**10.10.4.275 GenApi::Integer& EventCounter0StartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start [Event](#).

Visibility: Expert

**10.10.4.276 GenApi::Integer& EventCounter0StartTimestamp**

Description: Returns the Timestamp of the Counter 0 Start [Event](#).

Visibility: Expert

**10.10.4.277 GenApi::Integer& EventCounter1End**

Description: Returns the unique Identifier of the Counter 1 End type of [Event](#).

Visibility: Expert

**10.10.4.278 GenApi::Integer& EventCounter1EndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End [Event](#).

Visibility: Expert

**10.10.4.279 GenApi::Integer& EventCounter1EndTimestamp**

Description: Returns the Timestamp of the Counter 1 End [Event](#).

Visibility: Expert

**10.10.4.280 GenApi::Integer& EventCounter1Start**

Description: Returns the unique Identifier of the Counter 1 Start type of [Event](#).

Visibility: Expert

**10.10.4.281 GenApi::Integer& EventCounter1StartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start [Event](#).

Visibility: Expert

**10.10.4.282 GenApi::Integer& EventCounter1StartTimestamp**

Description: Returns the Timestamp of the Counter 1 Start [Event](#).

Visibility: Expert

**10.10.4.283 GenApi::Integer& EventEncoder0Restarted**

Description: Returns the unique Identifier of the Encoder 0 Restarted type of [Event](#).

Visibility: Expert

**10.10.4.284 GenApi::Integer& EventEncoder0RestartedFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted [Event](#).

Visibility: Expert

**10.10.4.285 GenApi::Integer& EventEncoder0RestartedTimestamp**

Description: Returns the Timestamp of the Encoder 0 Restarted [Event](#).

Visibility: Expert

**10.10.4.286 GenApi::Integer& EventEncoder0Stopped**

Description: Returns the unique Identifier of the Encoder 0 Stopped type of [Event](#).

Visibility: Expert

**10.10.4.287 GenApi::Integer& EventEncoder0StoppedFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped [Event](#).

Visibility: Expert

**10.10.4.288 GenApi::Integer& EventEncoder0StoppedTimestamp**

Description: Returns the Timestamp of the Encoder 0 Stopped [Event](#).

Visibility: Expert

**10.10.4.289 GenApi::Integer& EventEncoder1Restarted**

Description: Returns the unique Identifier of the Encoder 1 Restarted type of [Event](#).

Visibility: Expert

**10.10.4.290 GenApi::Integer& EventEncoder1RestartedFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted [Event](#).

Visibility: Expert

**10.10.4.291 GenApi::Integer& EventEncoder1RestartedTimestamp**

Description: Returns the Timestamp of the Encoder 1 Restarted [Event](#).

Visibility: Expert

**10.10.4.292 GenApi::Integer& EventEncoder1Stopped**

Description: Returns the unique Identifier of the Encoder 1 Stopped type of [Event](#).

Visibility: Expert

**10.10.4.293 GenApi::Integer& EventEncoder1StoppedFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped [Event](#).

Visibility: Expert

**10.10.4.294 GenApi::Integer& EventEncoder1StoppedTimestamp**

Description: Returns the Timestamp of the Encoder 1 Stopped [Event](#).

Visibility: Expert

**10.10.4.295 GenApi::Integer& EventError**

Description: Returns the unique identifier of the Error type of [Event](#).

Visibility:

**10.10.4.296 GenApi::Integer& EventErrorCode**

Description: Returns the error code for the error that happened Visibility:

**10.10.4.297 GenApi::Integer& EventErrorFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Error [Event](#).

Visibility:

**10.10.4.298 GenApi::Integer& EventErrorTimestamp**

Description: Returns the Timestamp of the Error [Event](#).

Visibility:

**10.10.4.299 GenApi::Integer& EventExposureEnd**

Description: Returns the unique identifier of the Exposure End type of [Event](#).

Visibility:

**10.10.4.300 GenApi::Integer& EventExposureEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End [Event](#).

Visibility:

**10.10.4.301 GenApi::Integer& EventExposureEndTimestamp**

Description: Returns the Timestamp of the Exposure End [Event](#).

Visibility:

**10.10.4.302 GenApi::Integer& EventExposureStart**

Description: Returns the unique Identifier of the Exposure Start type of [Event](#).

Visibility: Expert

**10.10.4.303 GenApi::Integer& EventExposureStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start [Event](#).

Visibility: Expert



**10.10.4.304 GenApi::Integer& EventExposureStartTimestamp**

Description: Returns the Timestamp of the Exposure Start [Event](#).

Visibility: Expert

**10.10.4.305 GenApi::Integer& EventFrameBurstEnd**

Description: Returns the unique Identifier of the Frame Burst End type of [Event](#).

Visibility: Expert

**10.10.4.306 GenApi::Integer& EventFrameBurstEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End [Event](#).

Visibility: Expert

**10.10.4.307 GenApi::Integer& EventFrameBurstEndTimestamp**

Description: Returns the Timestamp of the Frame Burst End [Event](#).

Visibility: Expert

**10.10.4.308 GenApi::Integer& EventFrameBurstStart**

Description: Returns the unique Identifier of the Frame Burst Start type of [Event](#).

Visibility: Expert

**10.10.4.309 GenApi::Integer& EventFrameBurstStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start [Event](#).

Visibility: Expert

**10.10.4.310 GenApi::Integer& EventFrameBurstStartTimestamp**

Description: Returns the Timestamp of the Frame Burst Start [Event](#).

Visibility: Expert

**10.10.4.311 GenApi::Integer& EventFrameEnd**

Description: Returns the unique Identifier of the Frame End type of [Event](#).

Visibility: Expert

**10.10.4.312 GenApi::Integer& EventFrameEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End [Event](#).

Visibility: Expert

**10.10.4.313 GenApi::Integer& EventFrameEndTimestamp**

Description: Returns the Timestamp of the Frame End [Event](#).

Visibility: Expert

**10.10.4.314 GenApi::Integer& EventFrameStart**

Description: Returns the unique Identifier of the Frame Start type of [Event](#).

Visibility: Expert

**10.10.4.315 GenApi::Integer& EventFrameStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start [Event](#).

Visibility: Expert

**10.10.4.316 GenApi::Integer& EventFrameStartTimestamp**

Description: Returns the Timestamp of the Frame Start [Event](#).

Visibility: Expert

**10.10.4.317 GenApi::Integer& EventFrameTransferEnd**

Description: Returns the unique Identifier of the Frame Transfer End type of [Event](#).

Visibility: Expert

**10.10.4.318 GenApi::Integer& EventFrameTransferEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End [Event](#).

Visibility: Expert

**10.10.4.319 GenApi::Integer& EventFrameTransferEndTimestamp**

Description: Returns the Timestamp of the Frame Transfer End [Event](#).

Visibility: Expert

**10.10.4.320 GenApi::Integer& EventFrameTransferStart**

Description: Returns the unique Identifier of the Frame Transfer Start type of [Event](#).

Visibility: Expert

**10.10.4.321 GenApi::Integer& EventFrameTransferStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start [Event](#).

Visibility: Expert

**10.10.4.322 GenApi::Integer& EventFrameTransferStartTimestamp**

Description: Returns the Timestamp of the Frame Transfer Start [Event](#).

Visibility: Expert

**10.10.4.323 GenApi::Integer& EventFrameTrigger**

Description: Returns the unique Identifier of the FrameTrigger type of [Event](#).

It can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type event received. Visibility: Expert

**10.10.4.324 GenApi::Integer& EventFrameTriggerFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger [Event](#).

Visibility: Expert

**10.10.4.325 GenApi::Integer& EventFrameTriggerTimestamp**

Description: Returns the Timestamp of the FrameTrigger [Event](#).

It can be used to determine precisely when the event occurred. Visibility: Expert

**10.10.4.326 GenApi::Integer& EventLine0AnyEdge**

Description: Returns the unique Identifier of the Line 0 Any Edge type of [Event](#).

Visibility: Expert

**10.10.4.327 GenApi::Integer& EventLine0AnyEdgeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge [Event](#).

Visibility: Expert

**10.10.4.328 GenApi::Integer& EventLine0AnyEdgeTimestamp**

Description: Returns the Timestamp of the Line 0 Any Edge [Event](#).

Visibility: Expert

**10.10.4.329 GenApi::Integer& EventLine0FallingEdge**

Description: Returns the unique Identifier of the Line 0 Falling Edge type of [Event](#).

Visibility: Expert

**10.10.4.330 GenApi::Integer& EventLine0FallingEdgeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge [Event](#).

Visibility: Expert

**10.10.4.331 GenApi::Integer& EventLine0FallingEdgeTimestamp**

Description: Returns the Timestamp of the Line 0 Falling Edge [Event](#).

Visibility: Expert

**10.10.4.332 GenApi::Integer& EventLine0RisingEdge**

Description: Returns the unique Identifier of the Line 0 Rising Edge type of [Event](#).

Visibility: Expert

**10.10.4.333 GenApi::Integer& EventLine0RisingEdgeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge [Event](#).

Visibility: Expert

**10.10.4.334 GenApi::Integer& EventLine0RisingEdgeTimestamp**

Description: Returns the Timestamp of the Line 0 Rising Edge [Event](#).

Visibility: Expert

**10.10.4.335 GenApi::Integer& EventLine1AnyEdge**

Description: Returns the unique Identifier of the Line 1 Any Edge type of [Event](#).

Visibility: Expert

**10.10.4.336 GenApi::Integer& EventLine1AnyEdgeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge [Event](#).

Visibility: Expert

**10.10.4.337 GenApi::Integer& EventLine1AnyEdgeTimestamp**

Description: Returns the Timestamp of the Line 1 Any Edge [Event](#).

Visibility: Expert

**10.10.4.338 GenApi::Integer& EventLine1FallingEdge**

Description: Returns the unique Identifier of the Line 1 Falling Edge type of [Event](#).

Visibility: Expert

**10.10.4.339 GenApi::Integer& EventLine1FallingEdgeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge [Event](#).

Visibility: Expert

**10.10.4.340 GenApi::Integer& EventLine1FallingEdgeTimestamp**

Description: Returns the Timestamp of the Line 1 Falling Edge [Event](#).

Visibility: Expert

**10.10.4.341 GenApi::Integer& EventLine1RisingEdge**

Description: Returns the unique Identifier of the Line 1 Rising Edge type of [Event](#).

Visibility: Expert

**10.10.4.342 GenApi::Integer& EventLine1RisingEdgeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge [Event](#).

Visibility: Expert

**10.10.4.343 GenApi::Integer& EventLine1RisingEdgeTimestamp**

Description: Returns the Timestamp of the Line 1 Rising Edge [Event](#).

Visibility: Expert

**10.10.4.344 GenApi::Integer& EventLinkSpeedChange**

Description: Returns the unique Identifier of the Link Speed Change type of [Event](#).

Visibility: Expert

**10.10.4.345 GenApi::Integer& EventLinkSpeedChangeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change [Event](#).

Visibility: Expert

**10.10.4.346 GenApi::Integer& EventLinkSpeedChangeTimestamp**

Description: Returns the Timestamp of the Link Speed Change [Event](#).

Visibility: Expert

**10.10.4.347 GenApi::Integer& EventLinkTrigger0**

Description: Returns the unique Identifier of the Link Trigger 0 type of [Event](#).

Visibility: Expert

**10.10.4.348 GenApi::Integer& EventLinkTrigger0FrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 [Event](#).

Visibility: Expert

**10.10.4.349 GenApi::Integer& EventLinkTrigger0Timestamp**

Description: Returns the Timestamp of the Link Trigger 0 [Event](#).

Visibility: Expert

**10.10.4.350 GenApi::Integer& EventLinkTrigger1**

Description: Returns the unique Identifier of the Link Trigger 1 type of [Event](#).

Visibility: Expert

**10.10.4.351 GenApi::Integer& EventLinkTrigger1FrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 [Event](#).

Visibility: Expert

**10.10.4.352 GenApi::Integer& EventLinkTrigger1Timestamp**

Description: Returns the Timestamp of the Link Trigger 1 [Event](#).

Visibility: Expert

**10.10.4.353 GenApi::EnumerationT<EventNotificationEnums>& EventNotification**

Description: Enables/Disables the selected event.

Visibility:

**10.10.4.354 GenApi::EnumerationT<EventSelectorEnums>& EventSelector**

Description: Selects which [Event](#) to enable or disable.

Visibility:

**10.10.4.355 GenApi::Integer& EventSequencerSetChange**

Description: Returns the unique Identifier of the Sequencer Set Change type of [Event](#).

Visibility: Expert

**10.10.4.356 GenApi::Integer& EventSequencerSetChangeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change [Event](#).

Visibility: Expert

**10.10.4.357 GenApi::Integer& EventSequencerSetChangeTimestamp**

Description: Returns the Timestamp of the Sequencer Set Change [Event](#).

Visibility: Expert

**10.10.4.358 GenApi::IString& EventSerialData**

Description: Returns the serial data that was received.

Visibility:

**10.10.4.359 GenApi::Integer& EventSerialDataLength**

Description: Returns the length of the received serial data that was included in the event payload.

Visibility:

**10.10.4.360 GenApi::Integer& EventSerialPortReceive**

Description: Returns the unique identifier of the Serial Port Receive type of [Event](#).

Visibility:

**10.10.4.361 GenApi::Integer& EventSerialPortReceiveTimestamp**

Description: Returns the Timestamp of the Serial Port Receive [Event](#).

Visibility:

**10.10.4.362 GenApi::Boolean& EventSerialReceiveOverflow**

Description: Returns the status of the event serial receive overflow.

Visibility:

**10.10.4.363 GenApi::Integer& EventStream0TransferBlockEnd**

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of [Event](#).

Visibility: Expert

**10.10.4.364 GenApi::Integer& EventStream0TransferBlockEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End [Event](#).

Visibility: Expert

**10.10.4.365 GenApi::Integer& EventStream0TransferBlockEndTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Block End [Event](#).

Visibility: Expert



**10.10.4.366 GenApi::Integer& EventStream0TransferBlockStart**

Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of [Event](#).

Visibility: Expert

**10.10.4.367 GenApi::Integer& EventStream0TransferBlockStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start [Event](#).

Visibility: Expert

**10.10.4.368 GenApi::Integer& EventStream0TransferBlockStartTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Block Start [Event](#).

Visibility: Expert

**10.10.4.369 GenApi::Integer& EventStream0TransferBlockTrigger**

Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of [Event](#).

Visibility: Expert

**10.10.4.370 GenApi::Integer& EventStream0TransferBlockTriggerFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger [Event](#).

Visibility: Expert

**10.10.4.371 GenApi::Integer& EventStream0TransferBlockTriggerTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger [Event](#).

Visibility: Expert

**10.10.4.372 GenApi::Integer& EventStream0TransferBurstEnd**

Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of [Event](#).

Visibility: Expert

**10.10.4.373 GenApi::Integer& EventStream0TransferBurstEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End [Event](#).

Visibility: Expert

**10.10.4.374 GenApi::Integer& EventStream0TransferBurstEndTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Burst End [Event](#).

Visibility: Expert

**10.10.4.375 GenApi::Integer& EventStream0TransferBurstStart**

Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of [Event](#).

Visibility: Expert

**10.10.4.376 GenApi::Integer& EventStream0TransferBurstStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start [Event](#).

Visibility: Expert

**10.10.4.377 GenApi::Integer& EventStream0TransferBurstStartTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Burst Start [Event](#).

Visibility: Expert

**10.10.4.378 GenApi::Integer& EventStream0TransferEnd**

Description: Returns the unique Identifier of the Stream 0 Transfer End type of [Event](#).

Visibility: Expert

**10.10.4.379 GenApi::Integer& EventStream0TransferEndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End [Event](#).

Visibility: Expert

**10.10.4.380 GenApi::Integer& EventStream0TransferEndTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer End [Event](#).

Visibility: Expert

**10.10.4.381 GenApi::Integer& EventStream0TransferOverflow**

Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of [Event](#).

Visibility: Expert

**10.10.4.382 GenApi::Integer& EventStream0TransferOverflowFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow [Event](#).

Visibility: Expert

**10.10.4.383 GenApi::Integer& EventStream0TransferOverflowTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Overflow [Event](#).

Visibility: Expert

**10.10.4.384 GenApi::Integer& EventStream0TransferPause**

Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of [Event](#).

Visibility: Expert

**10.10.4.385 GenApi::Integer& EventStream0TransferPauseFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause [Event](#).

Visibility: Expert

**10.10.4.386 GenApi::Integer& EventStream0TransferPauseTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Pause [Event](#).

Visibility: Expert

**10.10.4.387 GenApi::Integer& EventStream0TransferResume**

Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of [Event](#).

Visibility: Expert

**10.10.4.388 GenApi::Integer& EventStream0TransferResumeFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume [Event](#).

Visibility: Expert

**10.10.4.389 GenApi::Integer& EventStream0TransferResumeTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Resume [Event](#).

Visibility: Expert

**10.10.4.390 GenApi::Integer& EventStream0TransferStart**

Description: Returns the unique Identifier of the Stream 0 Transfer Start type of [Event](#).

Visibility: Expert

**10.10.4.391 GenApi::Integer& EventStream0TransferStartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start [Event](#).

Visibility: Expert

**10.10.4.392 GenApi::Integer& EventStream0TransferStartTimestamp**

Description: Returns the Timestamp of the Stream 0 Transfer Start [Event](#).

Visibility: Expert

**10.10.4.393 GenApi::Integer& EventTest**

Description: Returns the unique identifier of the Test type of [Event](#).

Visibility:

**10.10.4.394 GenApi::Integer& EventTestTimestamp**

Description: Returns the Timestamp of the Test [Event](#).

Visibility:

**10.10.4.395 GenApi::Integer& EventTimer0End**

Description: Returns the unique Identifier of the Timer 0 End type of [Event](#).

Visibility: Expert

**10.10.4.396 GenApi::Integer& EventTimer0EndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End [Event](#).

Visibility: Expert

**10.10.4.397 GenApi::Integer& EventTimer0EndTimestamp**

Description: Returns the Timestamp of the Timer 0 End [Event](#).

Visibility: Expert

**10.10.4.398 GenApi::Integer& EventTimer0Start**

Description: Returns the unique Identifier of the Timer 0 Start type of [Event](#).

Visibility: Expert

**10.10.4.399 GenApi::Integer& EventTimer0StartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start [Event](#).

Visibility: Expert

**10.10.4.400 GenApi::Integer& EventTimer0StartTimestamp**

Description: Returns the Timestamp of the Timer 0 Start [Event](#).

Visibility: Expert

**10.10.4.401 GenApi::Integer& EventTimer1End**

Description: Returns the unique Identifier of the Timer 1 End type of [Event](#).

Visibility: Expert

**10.10.4.402 GenApi::Integer& EventTimer1EndFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End [Event](#).

Visibility: Expert

**10.10.4.403 GenApi::Integer& EventTimer1EndTimestamp**

Description: Returns the Timestamp of the Timer 1 End [Event](#).

Visibility: Expert

**10.10.4.404 GenApi::Integer& EventTimer1Start**

Description: Returns the unique Identifier of the Timer 1 Start type of [Event](#).

Visibility: Expert

**10.10.4.405 GenApi::Integer& EventTimer1StartFrameID**

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start [Event](#).

Visibility: Expert

**10.10.4.406 GenApi::Integer& EventTimer1StartTimestamp**

Description: Returns the Timestamp of the Timer 1 Start [Event](#).

Visibility: Expert

**10.10.4.407 GenApi::EnumerationT<ExposureActiveModeEnums>& ExposureActiveMode**

Description: Control sensor active exposure mode.

Visibility:

**10.10.4.408 GenApi::EnumerationT<ExposureAutoEnums>& ExposureAuto**

Description: Sets the automatic exposure mode Visibility:

**10.10.4.409 GenApi::EnumerationT<ExposureModeEnums>& ExposureMode**

Description:

Sets the operation mode of the Exposure.

Visibility:

**10.10.4.410 GenApi::IFloat& ExposureTime**

Description:

Exposure time in microseconds when Exposure Mode is Timed.

Visibility:

**10.10.4.411 GenApi::IEnumerationT<ExposureTimeModeEnums>& ExposureTimeMode**

Description: Sets the configuration mode of the ExposureTime feature.

Visibility: Beginner

**10.10.4.412 GenApi::IEnumerationT<ExposureTimeSelectorEnums>& ExposureTimeSelector**

Description: Selects which exposure time is controlled by the ExposureTime feature.

This allows for independent control over the exposure components. Visibility: Beginner

**10.10.4.413 GenApi::ICommand& FactoryReset**

Description: Returns all user tables to factory default Visibility:

**10.10.4.414 GenApi::IRegister& FileAccessBuffer**

Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.

Visibility: Guru

**10.10.4.415 GenApi::Integer& FileAccessLength**

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

**10.10.4.416 GenApi::Integer& FileAccessOffset**

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

**10.10.4.417 GenApi::IEnumerationT<FileOpenModeEnums>& FileOpenMode**

Description:

The mode of the file when it is opened.

The file can be opened for reading, writing or both. This must be set before opening the file.

Visibility:

**10.10.4.418 GenApi::ICommand& FileOperationExecute**

Description:

This is a command that executes the selected file operation on the selected file.

Visibility:

**10.10.4.419 GenApi::Integer& FileOperationResult**

Description: Represents the file operation result.

For Read or Write operations, the number of successfully read/written bytes is returned. Visibility:

**10.10.4.420 GenApi::IEnumerationT<FileOperationSelectorEnums>& FileOperationSelector**

Description:

Sets operation to execute on the selected file when the execute command is given.

Visibility:

**10.10.4.421 GenApi::IEnumerationT<FileOperationStatusEnums>& FileOperationStatus**

Description: Represents the file operation execution status.

Visibility:

**10.10.4.422 GenApi::IEnumerationT<FileSelectorEnums>& FileSelector**

Description:

Selects which file is being operated on.

This must be set before performing any file operations.

Visibility:



**10.10.4.423 GenApi::Integer& FileSize**

Description: Represents the size of the selected file in bytes.

Visibility:

**10.10.4.424 GenApi::IFloat& Gain**

Description:

Controls the amplification of the video signal in dB.

Visibility:

**10.10.4.425 GenApi::IEnumerationT<GainAutoEnums>& GainAuto**

Description:

Sets the automatic gain mode.

Set to Off for manual control. Set to Once for a single automatic adjustment then return to Off. Set to Continuous for constant adjustment. In automatic modes, the camera adjusts the gain to maximize the dynamic range. Visibility:

**10.10.4.426 GenApi::IEnumerationT<GainAutoBalanceEnums>& GainAutoBalance**

Description: Sets the mode for automatic gain balancing between the sensor color channels or taps.

The gain coefficients of each channel or tap are adjusted so they are matched. Visibility: Beginner

**10.10.4.427 GenApi::IEnumerationT<GainSelectorEnums>& GainSelector**

Description: Selects which gain to control.

The All selection is a total amplification across all channels (or taps).

Visibility:

**10.10.4.428 GenApi::IFloat& Gamma**

Description: Controls the gamma correction of pixel intensity.

Visibility:

**10.10.4.429 GenApi::IBoolean& GammaEnable**

Description: Enables/disables gamma correction.

Visibility:

**10.10.4.430 GenApi::Integer& GevActiveLinkCount**

Description: Indicates the current number of active logical links.

Visibility: Expert

**10.10.4.431 GenApi::IEnumerationT<GevCCPEnums>& GevCCP**

Description: Controls the device access privilege of an application.

Visibility:

**10.10.4.432 GenApi::Integer& GevCurrentDefaultGateway**

Description: Reports the default gateway IP address to be used on the given logical link.

Visibility:

**10.10.4.433 GenApi::Integer& GevCurrentIPAddress**

Description: Reports the IP address for the given logical link.

Visibility:

**10.10.4.434 GenApi::IBoolean& GevCurrentIPConfigurationDHCP**

Description: Controls whether the DHCP IP configuration scheme is activated on the given logical link.

Visibility:

**10.10.4.435 GenApi::IBoolean& GevCurrentIPConfigurationLLA**

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.

Visibility:

**10.10.4.436 GenApi::IBoolean& GevCurrentIPConfigurationPersistentIP**

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.

Visibility:

**10.10.4.437 GenApi::IEnumerationT<GevCurrentPhysicalLinkConfigurationEnums>& GevCurrentPhysicalLinkConfiguration**

Description: Indicates the current physical link configuration of the device.

Visibility: Expert

**10.10.4.438 GenApi::Integer& GevCurrentSubnetMask**

Description: Reports the subnet mask of the given logical link.

Visibility:

**10.10.4.439 GenApi::Integer& GevDiscoveryAckDelay**

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

Visibility: Expert

**10.10.4.440 GenApi::String& GevFirstURL**

Description: The first choice of URL for the XML device description file.

Visibility:

**10.10.4.441 GenApi::Boolean& GevGVCPExtendedStatusCodes**

Description: Enables the generation of extended status codes.

Visibility: Guru

**10.10.4.442 GenApi::EnumerationT<GevGVCPExtendedStatusCodesSelectorEnums>& GevGVCPExtendedStatusCodesSelector**

Description: Selects the GigE Vision version to control extended status codes for.

Visibility: Guru

**10.10.4.443 GenApi::Boolean& GevGVCPHeartbeatDisable**

Description: Disables the GVCP heartbeat.

Visibility:

**10.10.4.444 GenApi::Boolean& GevGVCPPendingAck**

Description: Enables the generation of PENDING\_ACK.

Visibility:

**10.10.4.445 GenApi::Integer& GevGVCPendingTimeout**

Description: Indicates the longest GVCP command execution time before the device returns a PENDING\_ACK in milliseconds.

Visibility:

**10.10.4.446 GenApi::IEnumerationT<GevGVSPExtendedIDModeEnums>& GevGVSPExtendedIDMode**

Description: Enables the extended IDs mode.

Visibility: Expert

**10.10.4.447 GenApi::Integer& GevHeartbeatTimeout**

Description: Indicates the current heartbeat timeout in milliseconds.

Visibility:

**10.10.4.448 GenApi::IBoolean& GevIEEE1588**

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.

Visibility:

**10.10.4.449 GenApi::IEnumerationT<GevIEEE1588ClockAccuracyEnums>& GevIEEE1588ClockAccuracy**

Description: Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.

Visibility:

**10.10.4.450 GenApi::IEnumerationT<GevIEEE1588ModeEnums>& GevIEEE1588Mode**

Description: Provides the mode of the IEEE 1588 clock.

Visibility:

**10.10.4.451 GenApi::IEnumerationT<GevIEEE1588StatusEnums>& GevIEEE1588Status**

Description: Provides the status of the IEEE 1588 clock.

Visibility:

**10.10.4.452 GenApi::Integer & GevInterfaceSelector**

Description: Selects which logical link to control.

Visibility:

**10.10.4.453 GenApi::EnumerationT<GevIPConfigurationStatusEnums> & GevIPConfigurationStatus**

Description: Reports the current IP configuration status.

Visibility: Beginner

**10.10.4.454 GenApi::Integer & GevMACAddress**

Description: MAC address of the logical link.

Visibility:

**10.10.4.455 GenApi::Integer & GevMCDA**

Description: Controls the destination IP address of the message channel Visibility:

**10.10.4.456 GenApi::Integer & GevMCPHostPort**

Description: The port to which the device must send messages Visibility:

**10.10.4.457 GenApi::Integer & GevMCRC**

Description: Indicates the number of retries of the message channel.

Visibility:

**10.10.4.458 GenApi::Integer & GevMCSP**

Description: Indicates the source port of the message channel.

Visibility:

**10.10.4.459 GenApi::Integer & GevMCTT**

Description: Indicates the transmission timeout of the message channel.

Visibility:

**10.10.4.460 GenApi::Integer& GevNumberOfInterfaces**

Description: Indicates the number of physical network interfaces supported by this device.

Visibility:

**10.10.4.461 GenApi::Boolean& GevPAUSEFrameReception**

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.

Visibility: Expert

**10.10.4.462 GenApi::Boolean& GevPAUSEFrameTransmission**

Description: Controls whether PAUSE Frames can be generated on the given logical link.

Visibility: Expert

**10.10.4.463 GenApi::Integer& GevPersistentDefaultGateway**

Description: Controls the persistent default gateway for this logical link.

Visibility:

**10.10.4.464 GenApi::Integer& GevPersistentIPAddress**

Description: Controls the Persistent IP address for this logical link.

Visibility:

**10.10.4.465 GenApi::Integer& GevPersistentSubnetMask**

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.

Visibility:

**10.10.4.466 GenApi::EnumerationT<GevPhysicalLinkConfigurationEnums>& GevPhysicalLinkConfiguration**

Description: Controls the principal physical link configuration to use on next restart/power-up of the device.

Visibility: Expert

**10.10.4.467 GenApi::Integer& GevPrimaryApplicationIPAddress**

Description: Returns the address of the primary application.

Visibility: Guru

**10.10.4.468 GenApi::Integer & GevPrimaryApplicationSocket**

Description: Returns the UDP source port of the primary application.

Visibility: Guru

**10.10.4.469 GenApi::Integer & GevPrimaryApplicationSwitchoverKey**

Description: Controls the key to use to authenticate primary application switchover requests.

Visibility: Guru

**10.10.4.470 GenApi::Boolean & GevSCCFGAllInTransmission**

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

Visibility: Guru

**10.10.4.471 GenApi::Boolean & GevSCCFGExtendedChunkData**

Description: Enables cameras to use the extended chunk data payload type for this stream channel.

Visibility:

**10.10.4.472 GenApi::Boolean & GevSCCFGPacketResendDestination**

Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.

When True, the source IP address provided in the packet resend command packet is used. When False, the value set in the GevSCDA[GevStreamChannelSelector] feature is used. Visibility: Guru

**10.10.4.473 GenApi::Boolean & GevSCCFGUnconditionalStreaming**

Description: Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).

Visibility:

**10.10.4.474 GenApi::Integer & GevSCDA**

Description: Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.

Visibility:

**10.10.4.475 GenApi::Integer&GevSCPD**

Description: Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.

This can be used as a crude flow-control mechanism if the application or the network infrastructure cannot keep up with the packets coming from the device. Visibility:

**10.10.4.476 GenApi::Integer&GevSCPDirection**

Description: Transmit or Receive of the channel Visibility:

**10.10.4.477 GenApi::Integer&GevSCPHostPort**

Description: Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.

Visibility:

**10.10.4.478 GenApi::Integer&GevSCPIInterfaceIndex**

Description: Index of the logical link to use.

Visibility:

**10.10.4.479 GenApi::Boolean&GevSCPSBigEndian**

Description: Endianess of multi-byte pixel data for this stream.

Visibility:

**10.10.4.480 GenApi::Boolean&GevSCPSDoNotFragment**

Description: The state of this feature is copied into the "do not fragment" bit of the IP header of each stream packet.

Visibility:

**10.10.4.481 GenApi::Boolean&GevSCPSFireTestPacket**

Description: Sends a test packet.

Visibility:



**10.10.4.482 GenApi::Integer& GevSCSPacketSize**

Description: Specifies the stream packet size (in bytes) to send on this channel.

Visibility:

**10.10.4.483 GenApi::Integer& GevSCSP**

Description: Indicates the source port of the stream channel.

Visibility:

**10.10.4.484 GenApi::Boolean& GevSCZoneConfigurationLock**

Description: Controls whether the selected stream channel multi-zone configuration is locked.

When locked, the GVSP transmitter is not allowed to change the number of zones and their direction during block acquisition and transmission. Visibility: Guru

**10.10.4.485 GenApi::Integer& GevSCZoneCount**

Description: Reports the number of zones per block transmitted on the selected stream channel.

Visibility: Guru

**10.10.4.486 GenApi::Integer& GevSCZoneDirectionAll**

Description: Reports the transmission direction of each zone transmitted on the selected stream channel.

Visibility: Guru

**10.10.4.487 GenApi::String& GevSecondURL**

Description: The second choice of URL to the XML device description file.

Visibility:

**10.10.4.488 GenApi::Integer& GevStreamChannelSelector**

Description: Selects the stream channel to control.

Visibility:

**10.10.4.489 GenApi::IBoolean& GevSupportedOption**

Description: Returns if the selected GEV option is supported.

Visibility:

**10.10.4.490 GenApi::IEnumerationT<GevSupportedOptionSelectorEnums>& GevSupportedOptionSelector**

Description: Selects the GEV option to interrogate for existing support.

Visibility:

**10.10.4.491 GenApi::IInteger& GevTimestampTickFrequency**

Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).

Visibility:

**10.10.4.492 GenApi::IInteger& GuiXmlManifestAddress**

Description: Location of the GUI XML manifest table.

Visibility:

**10.10.4.493 GenApi::IInteger& Height**

Description:

Height of the image provided by the device (in pixels).

Visibility:

**10.10.4.494 GenApi::IInteger& HeightMax**

Description: Maximum height of the image (in pixels).

This dimension is calculated after vertical binning. HeightMax does not take into account the current Region of interest (Height or OffsetY). Visibility:

**10.10.4.495 GenApi::IBoolean& ImageComponentEnable**

Description: Controls if the selected component streaming is active.

Visibility: Beginner

**10.10.4.496 GenApi::IEnumerationT<ImageComponentSelectorEnums>& ImageComponentSelector**

Description: Selects a component to activate data streaming from.

Visibility: Beginner

**10.10.4.497 GenApi::IFloat& ImageCompressionBitrate**

Description: Control the rate of the produced compressed stream.

Visibility: Expert

**10.10.4.498 GenApi::IEnumerationT<ImageCompressionJPEGFormatOptionEnums>& ImageCompressionJPEGFormatOption**

Description: When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

Visibility: Expert

**10.10.4.499 GenApi::IEnumerationT<ImageCompressionModeEnums>& ImageCompressionMode**

Description: Visibility:

**10.10.4.500 GenApi::Integer& ImageCompressionQuality**

Description: Control the quality of the produced compressed stream.

Visibility: Expert

**10.10.4.501 GenApi::IEnumerationT<ImageCompressionRateOptionEnums>& ImageCompressionRateOption**

Description: Two rate controlling options are offered: fixed bit rate or fixed quality.

The exact implementation to achieve one or the other is vendor-specific. Visibility: Expert

**10.10.4.502 GenApi::IBoolean& IspEnable**

Description:

Controls whether the image processing core is used for optional pixel format mode (i.e. mono).

Visibility:

**10.10.4.503 GenApi::IFloat & LineFilterWidth**

Description: Filter width in microseconds for the selected line and filter combination Visibility:

**10.10.4.504 GenApi::IEnumerationT<LineFormatEnums> & LineFormat**

Description: Displays the current electrical format of the selected physical input or output Line.

Visibility:

**10.10.4.505 GenApi::IEnumerationT<LineInputFilterSelectorEnums> & LineInputFilterSelector**

Description: Selects the kind of input filter to configure: Deglitch or Debounce.

Visibility:

**10.10.4.506 GenApi::IBoolean & LineInverter**

Description: Controls the inversion of the signal of the selected input or output line.

Visibility:

**10.10.4.507 GenApi::IEnumerationT<LineModeEnums> & LineMode**

Description: Controls if the physical Line is used to Input or Output a signal.

Visibility:

**10.10.4.508 GenApi::Integer & LinePitch**

Description: Total number of bytes between 2 successive lines.

This feature is used to facilitate alignment of image data. Visibility: Expert

**10.10.4.509 GenApi::IEnumerationT<LineSelectorEnums> & LineSelector**

Description: Selects the physical line (or pin) of the external device connector to configure Visibility:

**10.10.4.510 GenApi::IEnumerationT<LineSourceEnums> & LineSource**

Description: Selects which internal acquisition or I/O source signal to output on the selected line.

LineMode must be Output. Visibility:

**10.10.4.511 GenApi::IBoolean& LineStatus**

Description: Returns the current status of the selected input or output Line Visibility:

**10.10.4.512 GenApi::Integer& LineStatusAll**

Description: Returns the current status of all the line status bits in a hexadecimal representation (Line 0 status corresponds to bit 0, Line 1 status with bit 1, etc).

This allows simultaneous reading of all line statuses at once. Visibility:

**10.10.4.513 GenApi::Integer& LinkErrorCount**

Description: Counts the number of error on the link.

Visibility:

**10.10.4.514 GenApi::Integer& LinkUptime**

Description: Time since the last phy negotiation (enumeration).

Visibility:

**10.10.4.515 GenApi::EnumerationT<LogicBlockLUTInputActivationEnums>& LogicBlockLUTInputActivation**

Description: Selects the activation mode of the Logic Input Source signal.

Visibility:

**10.10.4.516 GenApi::EnumerationT<LogicBlockLUTInputSelectorEnums>& LogicBlockLUTInputSelector**

Description: Controls which LogicBlockLUT Input Source & Activation to access.

Visibility:

**10.10.4.517 GenApi::EnumerationT<LogicBlockLUTInputSourceEnums>& LogicBlockLUTInputSource**

Description: Selects the source for the input into the Logic LUT.

Visibility:

**10.10.4.518 GenApi::IBoolean& LogicBlockLUTOutputValue**

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.

Visibility:

**10.10.4.519 GenApi::Integer & LogicBlockLUTOutputValueAll**

Description: Sets the value of all the output bits in the selected LUT.

Visibility:

**10.10.4.520 GenApi::Integer & LogicBlockLUTRowIndex**

Description: Controls the row of the truth table to access in the selected LUT.

Visibility:

**10.10.4.521 GenApi::EnumerationT<LogicBlockLUTSelectorEnums> & LogicBlockLUTSelector**

Description: Selects which LogicBlock LUT to configure Visibility:

**10.10.4.522 GenApi::EnumerationT<LogicBlockSelectorEnums> & LogicBlockSelector**

Description: Selects which LogicBlock to configure Visibility:

**10.10.4.523 GenApi::Boolean & LUTEnable**

Description:

Activates the selected LUT.

Visibility:

**10.10.4.524 GenApi::Integer & LUTIndex**

Description:

Control the index (offset) of the coefficient to access in the selected LUT.

Visibility:

**10.10.4.525 GenApi::EnumerationT<LUTSelectorEnums> & LUTSelector**

Description:

Selects which LUT to control.

Visibility:

**10.10.4.526 GenApi::Integer& LUTValue**

Description:

Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.

Visibility:

**10.10.4.527 GenApi::Integer& LUTValueAll**

Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.

Visibility: Guru

**10.10.4.528 GenApi::Integer& MaxDeviceResetTime**

Description: Time to wait until device reset complete (ms).

Visibility:

**10.10.4.529 GenApi::Integer& OffsetX**

Description:

Horizontal offset from the origin to the ROI (in pixels).

Visibility:

**10.10.4.530 GenApi::Integer& OffsetY**

Description:

Vertical offset from the origin to the ROI (in pixels).

Visibility:

**10.10.4.531 GenApi::Integer& PacketResendRequestCount**

Description: Counts the number of resend requests received from the host.

Visibility:

**10.10.4.532 GenApi::Integer& PayloadSize**

Description: Provides the number of bytes transferred for each image or chunk on the stream channel.

Visibility:

**10.10.4.533 GenApi::IEnumerationT<PixelColorFilterEnums>& PixelColorFilter**

Description: Type of color filter that is applied to the image.

Only applies to Bayer pixel formats. All others have no color filter.

Visibility:

**10.10.4.534 GenApi::Integer& PixelDynamicRangeMax**

Description: Maximum value that can be returned during the digitization process.

This corresponds to the brightest value of the camera. For color cameras, this returns the biggest value that each color component can take.

Visibility:

**10.10.4.535 GenApi::Integer& PixelDynamicRangeMin**

Description: Minimum value that can be returned during the digitization process.

This corresponds to the darkest value of the camera. For color cameras, this returns the smallest value that each color component can take.

Visibility:

**10.10.4.536 GenApi::IEnumerationT<PixelFormatEnums>& PixelFormat**

Description: Format of the pixel provided by the camera.

Visibility:

**10.10.4.537 GenApi::Integer& PixelFormatInfoID**

Description: Returns the value used by the streaming channels to identify the selected pixel format.

Visibility: Guru

**10.10.4.538 GenApi::IEnumerationT<PixelFormatInfoSelectorEnums>& PixelFormatInfoSelector**

Description: Select the pixel format for which the information will be returned.

Visibility: Guru

**10.10.4.539 GenApi::IEnumerationT<PixelSizeEnums>& PixelSize**

Description: Total size in bits of a pixel of the image.

Visibility:



**10.10.4.540 GenApi::IFloat& PowerSupplyCurrent**

Description:

Indicates the output current of the selected power supply (A).

Visibility:

**10.10.4.541 GenApi::IFloat& PowerSupplyVoltage**

Description:

Indicates the current voltage of the selected power supply (V).

Visibility:

**10.10.4.542 GenApi::IEnumerationT<RegionDestinationEnums>& RegionDestination**

Description: Control the destination of the selected region.

Visibility: Expert

**10.10.4.543 GenApi::IEnumerationT<RegionModeEnums>& RegionMode**

Description: Controls if the selected Region of interest is active and streaming.

Visibility: Beginner

**10.10.4.544 GenApi::IEnumerationT<RegionSelectorEnums>& RegionSelector**

Description: Selects the Region of interest to control.

The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently. Visibility: Beginner

**10.10.4.545 GenApi::IBoolean& ReverseX**

Description: Horizontally flips the image sent by the device.

The region of interest is applied after flipping. For color cameras the bayer pixel format is affected. For example, BayerRG16 changes to BayerGR16.

Visibility:

**10.10.4.546 GenApi::IBoolean& ReverseY**

Description: Vertically flips the image sent by the device.

The region of interest is applied after flipping. For color cameras the bayer pixel format is affected. For example, BayerRG16 changes to BayerGB16.

Visibility:

**10.10.4.547 GenApi::IEnumerationT<RgbTransformLightSourceEnums>& RgbTransformLightSource**

Description:

Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources.

Selecting a value also sets the white balance ratios (BalanceRatioRed and BalanceRatioBlue), but those can be overwritten through manual or auto white balance.

Visibility:

**10.10.4.548 GenApi::IFloat& Saturation**

Description: Controls the color saturation.

Visibility:

**10.10.4.549 GenApi::IBoolean& SaturationEnable**

Description: Enables/disables Saturation adjustment.

Visibility:

**10.10.4.550 GenApi::IFloat& Scan3dAxisMax**

Description: Maximum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

**10.10.4.551 GenApi::IFloat& Scan3dAxisMin**

Description: Minimum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

**10.10.4.552 GenApi::IFloat& Scan3dCoordinateOffset**

Description: Offset when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

**10.10.4.553 GenApi::IEnumerationT<Scan3dCoordinateReferenceSelectorEnums>& Scan3dCoordinateReferenceSelector**

Description: Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.

Visibility: Expert

**10.10.4.554 GenApi::IFloat& Scan3dCoordinateReferenceValue**

Description: Returns the reference value selected.

Reads the value of a rotation or translation value for the current (Anchor or Transformed) coordinate system transformation to the Reference system. Visibility: Expert

**10.10.4.555 GenApi::IFloat& Scan3dCoordinateScale**

Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

**10.10.4.556 GenApi::IEnumerationT<Scan3dCoordinateSelectorEnums>& Scan3dCoordinateSelector**

Description: Selects the individual coordinates in the vectors for 3D information/transformation.

Visibility: Expert

**10.10.4.557 GenApi::IEnumerationT<Scan3dCoordinateSystemEnums>& Scan3dCoordinateSystem**

Description: Specifies the Coordinate system to use for the device.

Visibility: Beginner

**10.10.4.558 GenApi::IEnumerationT<Scan3dCoordinateSystemReferenceEnums>& Scan3dCoordinateSystemReference**

Description: Defines coordinate system reference location.

Visibility: Expert

**10.10.4.559 GenApi::IEnumerationT<Scan3dCoordinateTransformSelectorEnums>& Scan3dCoordinateTransformSelector**

Description: Sets the index to read/write a coordinate transform value.

Visibility: Expert

**10.10.4.560 GenApi::IEnumerationT<Scan3dDistanceUnitEnums>& Scan3dDistanceUnit**

Description: Specifies the unit used when delivering calibrated distance data.

Visibility: Beginner

**10.10.4.561 GenApi::IBoolean& Scan3dInvalidDataFlag**

Description: Enables the definition of a non-valid flag value in the data stream.

Note that the confidence output is an alternate recommended way to identify non-valid pixels. Using an Scan3dInvalidDataValue may give processing penalties due to special handling. Visibility: Expert

**10.10.4.562 GenApi::IFloat& Scan3dInvalidDataValue**

Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.

Visibility: Expert

**10.10.4.563 GenApi::IEnumerationT<Scan3dOutputModeEnums>& Scan3dOutputMode**

Description: Controls the Calibration and data organization of the device, naming the coordinates transmitted.

Visibility: Expert

**10.10.4.564 GenApi::IFloat& Scan3dTransformValue**

Description: Specifies the transform value selected.

For translations (Scan3dCoordinateTransformSelector = TranslationX/Y/Z) it is expressed in the distance unit of the system, for rotations (Scan3dCoordinateTransformSelector =RotationX/Y/Z) in degrees. Visibility: Expert

**10.10.4.565 GenApi::IString& SensorDescription**

Description: Returns Sensor Description Visibility:

**10.10.4.566 GenApi::IEnumerationT<SensorDigitizationTapsEnums> & SensorDigitizationTaps**

Description: Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

Visibility: Expert

**10.10.4.567 GenApi::IInteger & SensorHeight**

Description: Effective height of the sensor in pixels.

Visibility:

**10.10.4.568 GenApi::IEnumerationT<SensorShutterModeEnums> & SensorShutterMode**

Description: Sets the shutter mode of the device.

Visibility:

**10.10.4.569 GenApi::IEnumerationT<SensorTapsEnums> & SensorTaps**

Description: Number of taps of the camera sensor.

Visibility: Expert

**10.10.4.570 GenApi::IInteger & SensorWidth**

Description: Effective width of the sensor in pixels.

Visibility:

**10.10.4.571 GenApi::IEnumerationT<SequencerConfigurationModeEnums> & SequencerConfigurationMode**

Description:

Controls whether or not a sequencer is in configuration mode.

Visibility:

**10.10.4.572 GenApi::IEnumerationT<SequencerConfigurationValidEnums> & SequencerConfigurationValid**

Description:

Display whether the current sequencer configuration is valid to run.

Visibility:

**10.10.4.573 GenApi::IBoolean& SequencerFeatureEnable**

Description:

Enables the selected feature and makes it active in all sequencer sets.

Visibility:

**10.10.4.574 GenApi::IEnumerationT<SequencerModeEnums>& SequencerMode**

Description: Controls whether or not a sequencer is active.

Visibility:

**10.10.4.575 GenApi::IInteger& SequencerPathSelector**

Description:

Selects branching path to be used for subsequent settings.

Visibility:

**10.10.4.576 GenApi::IInteger& SequencerSetActive**

Description: Displays the currently active sequencer set.

Visibility:

**10.10.4.577 GenApi::ICommand& SequencerSetLoad**

Description:

Loads currently selected sequencer to the current device configuration.

Visibility:

**10.10.4.578 GenApi::IInteger& SequencerSetNext**

Description: Specifies the next sequencer set.

Visibility:

**10.10.4.579 GenApi::ICommand& SequencerSetSave**

Description:

Saves the current device configuration to the currently selected sequencer set.

Visibility:

**10.10.4.580 GenApi::Integer& SequencerSetSelector**

Description:

Selects the sequencer set to which subsequent settings apply.

Visibility:

**10.10.4.581 GenApi::Integer& SequencerSetStart**

Description: Sets the first sequencer set to be used.

Visibility:

**10.10.4.582 GenApi::IEnumerationT<SequencerSetValidEnums>& SequencerSetValid**

Description:

Displays whether the currently selected sequencer set's register contents are valid to use.

Visibility:

**10.10.4.583 GenApi::IEnumerationT<SequencerTriggerActivationEnums>& SequencerTriggerActivation**

Description:

Specifies the activation mode of the sequencer trigger.

Visibility:

**10.10.4.584 GenApi::IEnumerationT<SequencerTriggerSourceEnums>& SequencerTriggerSource**

Description:

Specifies the internal signal or physical input line to use as the sequencer trigger source.

Visibility:

**10.10.4.585 GenApi::IEnumerationT<SerialPortBaudRateEnums>& SerialPortBaudRate**

Description: This feature controls the baud rate used by the selected serial port.

Visibility:

**10.10.4.586 GenApi::Integer& SerialPortDataBits**

Description: This feature controls the number of data bits used by the selected serial port.

Possible values that can be used are between 5 and 9. Visibility:

**10.10.4.587 GenApi::IEnumerationT<SerialPortParityEnums>& SerialPortParity**

Description: This feature controls the parity used by the selected serial port.

Visibility:

**10.10.4.588 GenApi::IEnumerationT<SerialPortSelectorEnums>& SerialPortSelector**

Description: Selects which serial port of the device to control.

Visibility:

**10.10.4.589 GenApi::IEnumerationT<SerialPortSourceEnums>& SerialPortSource**

Description: Specifies the physical input Line on which to receive serial data.

Visibility:

**10.10.4.590 GenApi::IEnumerationT<SerialPortStopBitsEnums>& SerialPortStopBits**

Description: This feature controls the number of stop bits used by the selected serial port.

Visibility:

**10.10.4.591 GenApi::Integer& SerialReceiveFramingErrorCount**

Description: Returns the number of framing errors that have occurred on the serial port.

Visibility:

**10.10.4.592 GenApi::Integer& SerialReceiveParityErrorCount**

Description: Returns the number of parity errors that have occurred on the serial port.

Visibility:



**10.10.4.593 GenApi::ICommand& SerialReceiveQueueClear**

Description: This is a command that clears the device serial port receive queue.

Visibility:

**10.10.4.594 GenApi::Integer& SerialReceiveQueueCurrentCharacterCount**

Description: Returns the number of characters currently in the serial port receive queue.

Visibility:

**10.10.4.595 GenApi::Integer& SerialReceiveQueueMaxCharacterCount**

Description: >Returns the maximum number of characters in the serial port receive queue.

Visibility:

**10.10.4.596 GenApi::Integer& SerialTransmitQueueCurrentCharacterCount**

Description: Returns the number of characters currently in the serial port transmit queue.

Visibility:

**10.10.4.597 GenApi::Integer& SerialTransmitQueueMaxCharacterCount**

Description: >Returns the maximum number of characters in the serial port transmit queue.

Visibility:

**10.10.4.598 GenApi::IFloat& Sharpening**

Description:

Controls the amount to sharpen a signal.

The sharpened amount is proportional to the difference between a pixel and its neighbors. A negative value smooths out the difference, while a positive value amplifies the difference. You can boost by a maximum of 8x, but smoothing is limited to 1x (in float). Default value: 2.0

Visibility:

#### 10.10.4.599 GenApi::IBoolean& SharpeningAuto

Description:

Enables/disables the auto sharpening feature.

When enabled, the camera automatically determines the sharpening threshold based on the noise level of the camera.

Visibility:

#### 10.10.4.600 GenApi::IBoolean& SharpeningEnable

Description:

Enables/disables the sharpening feature.

Sharpening is disabled by default.

Visibility:

#### 10.10.4.601 GenApi::IFloat& SharpeningThreshold

Description:

Controls the minimum intensity gradient change to invoke sharpening.

When "Sharpening Auto" is enabled, this is determined automatically by the device. The threshold is specified as a fraction of the total intensity range, and ranges from 0 to 0.25. A threshold higher than 25% produces little to no difference than 25%. High thresholds sharpen only areas with significant intensity changes. Low thresholds sharpen more areas.

Visibility:

#### 10.10.4.602 GenApi::ICommand& SoftwareSignalPulse

Description: Generates a pulse signal that can be used as a software trigger.

This command can be used to trigger other modules that accept a SoftwareSignal as trigger source. Visibility: Beginner

#### 10.10.4.603 GenApi::IEnumerationT<SoftwareSignalSelectorEnums>& SoftwareSignalSelector

Description: Selects which Software Signal features to control.

Visibility: Beginner

**10.10.4.604 GenApi::Integer& SourceCount**

Description: Controls or returns the number of sources supported by the device.

Visibility: Beginner

**10.10.4.605 GenApi::IEnumerationT<SourceSelectorEnums>& SourceSelector**

Description: Selects the source to control.

Visibility: Beginner

**10.10.4.606 GenApi::Integer& Test0001**

Description: For testing only.

Visibility:

**10.10.4.607 GenApi::ICommand& TestEventGenerate**

Description: This command generates a test event and sends it to the host.

Visibility:

**10.10.4.608 GenApi::IEnumerationT<TestPatternEnums>& TestPattern**

Description:

Selects the type of test pattern that is generated by the device as image source.

Visibility:

**10.10.4.609 GenApi::IEnumerationT<TestPatternGeneratorSelectorEnums>& TestPatternGeneratorSelector**

Description:

Selects which test pattern generator is controlled by the TestPattern feature.

Visibility:

**10.10.4.610 GenApi::Integer& TestPendingAck**

Description: Tests the device's pending acknowledge feature.

When this feature is written, the device waits a time period corresponding to the value of TestPendingAck before acknowledging the write. Visibility: Guru

**10.10.4.611 GenApi::IFloat& TimerDelay**

Description: Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer.

Visibility: Expert

**10.10.4.612 GenApi::IFloat& TimerDuration**

Description: Sets the duration (in microseconds) of the Timer pulse.

Visibility: Expert

**10.10.4.613 GenApi::ICommand& TimerReset**

Description: Does a software reset of the selected timer and starts it.

The timer starts immediately after the reset unless a timer trigger is active. Visibility: Expert

**10.10.4.614 GenApi::IEnumerationT<TimerSelectorEnums>& TimerSelector**

Description: Selects which Timer to configure.

Visibility: Expert

**10.10.4.615 GenApi::IEnumerationT<TimerStatusEnums>& TimerStatus**

Description: Returns the current status of the Timer.

Visibility: Expert

**10.10.4.616 GenApi::IEnumerationT<TimerTriggerActivationEnums>& TimerTriggerActivation**

Description: Selects the activation mode of the trigger to start the Timer.

Visibility: Expert

**10.10.4.617 GenApi::IEnumerationT<TimerTriggerSourceEnums>& TimerTriggerSource**

Description: Selects the source of the trigger to start the Timer.

Visibility: Expert

**10.10.4.618 GenApi::IFloat& TimerValue**

Description: Reads or writes the current value (in microseconds) of the selected Timer.

Visibility: Expert

**10.10.4.619 GenApi::Integer& Timestamp**

Description: Reports the current value of the device timestamp counter.

Visibility: Expert

**10.10.4.620 GenApi::ICommand& TimestampLatch**

Description: Latches the current timestamp counter into TimestampLatchValue.

Visibility:

**10.10.4.621 GenApi::Integer& TimestampLatchValue**

Description: Returns the latched value of the timestamp counter.

Visibility:

**10.10.4.622 GenApi::ICommand& TimestampReset**

Description: Resets the current value of the device timestamp counter.

Visibility:

**10.10.4.623 GenApi::Integer& TLParamsLocked**

Description: Visibility:

**10.10.4.624 GenApi::ICommand& TransferAbort**

Description: Aborts immediately the streaming of data block(s).

Aborting the transfer will result in the lost of the data that is present or currently entering in the block queue. However, the next new block received will be stored in the queue and transferred to the host when the streaming is restarted. If implemented, this feature should be available when the TransferControlMode is set to "UserControlled". Visibility: Expert

**10.10.4.625 GenApi::Integer& TransferBlockCount**

Description: Specifies the number of data blocks (images) that the device should stream before stopping.

This feature is only active if the Transfer Operation Mode is set to Multi Block. Visibility:

**10.10.4.626 GenApi::Integer& TransferBurstCount**

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.

Visibility: Expert

**10.10.4.627 GenApi::EnumerationT<TransferComponentSelectorEnums>& TransferComponentSelector**

Description: Selects the color component for the control of the TransferStreamChannel feature.

Visibility: Guru

**10.10.4.628 GenApi::EnumerationT<TransferControlModeEnums>& TransferControlMode**

Description: Selects the control method for the transfers.

Basic and Automatic start transmitting data as soon as there is enough data to fill a link layer packet. User Controlled allows you to directly control the transfer of blocks. Visibility:

**10.10.4.629 GenApi::EnumerationT<TransferOperationModeEnums>& TransferOperationMode**

Description: Selects the operation mode of the transfer.

Continuous is similar to Basic/Automatic but you can start/stop the transfer while acquisition runs independently. Multi Block transmits a specified number of blocks and then stops. Visibility:

**10.10.4.630 GenApi::ICommand& TransferPause**

Description: Pauses the streaming of data Block(s).

Pausing the streaming will immediately suspend the ongoing data transfer even if a block is partially transferred. The device will resume its transmission at the reception of a TransferResume command. Visibility: Guru

**10.10.4.631 GenApi::Integer& TransferQueueCurrentBlockCount**

Description: Returns number of data blocks (images) currently in the transfer queue.

Visibility:

**10.10.4.632 GenApi::Integer& TransferQueueMaxBlockCount**

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

**10.10.4.633 GenApi::EnumerationT<TransferQueueModeEnums>& TransferQueueMode**

Description: Specifies the operation mode of the transfer queue.

Visibility:

**10.10.4.634 GenApi::Integer& TransferQueueOverflowCount**

Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.

Visibility:

**10.10.4.635 GenApi::ICommand& TransferResume**

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.

Visibility: Guru

**10.10.4.636 GenApi::EnumerationT<TransferSelectorEnums>& TransferSelector**

Description: Selects which stream transfers are currently controlled by the selected Transfer features.

Visibility: Expert

**10.10.4.637 GenApi::ICommand& TransferStart**

Description: Starts the streaming of data blocks (images) out of the device.

This feature is available when the Transfer Control Mode is set to User Controlled. Visibility:

**10.10.4.638 GenApi::Boolean& TransferStatus**

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.

Visibility: Guru

**10.10.4.639 GenApi::EnumerationT<TransferStatusSelectorEnums>& TransferStatusSelector**

Description: Selects which status of the transfer module to read.

Visibility: Guru

**10.10.4.640 GenApi::ICommand& TransferStop**

Description: Stops the streaming of data block (images).

The current block transmission is completed. This feature is available when the Transfer Control Mode is set to User Controlled. Visibility:

**10.10.4.641 GenApi::Integer& TransferStreamChannel**

Description: Selects the streaming channel that will be used to transfer the selected stream of data.

In general, this feature can be omitted and the default streaming channel will be used. Visibility: Guru

**10.10.4.642 GenApi::IEnumerationT<TransferTriggerActivationEnums>& TransferTriggerActivation**

Description: Specifies the activation mode of the transfer control trigger.

Visibility: Guru

**10.10.4.643 GenApi::IEnumerationT<TransferTriggerModeEnums>& TransferTriggerMode**

Description: Controls if the selected trigger is active.

Visibility: Guru

**10.10.4.644 GenApi::IEnumerationT<TransferTriggerSelectorEnums>& TransferTriggerSelector**

Description: Selects the type of transfer trigger to configure.

Visibility: Guru

**10.10.4.645 GenApi::IEnumerationT<TransferTriggerSourceEnums>& TransferTriggerSource**

Description: Specifies the signal to use as the trigger source for transfers.

Visibility: Guru

**10.10.4.646 GenApi::IEnumerationT<TriggerActivationEnums>& TriggerActivation**

Description: Specifies the activation mode of the trigger.

Visibility:



**10.10.4.647 GenApi::IFloat& TriggerDelay**

Description:

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

Visibility:

**10.10.4.648 GenApi::Integer& TriggerDivider**

Description: Specifies a division factor for the incoming trigger pulses.

Visibility: Expert

**10.10.4.649 GenApi::ICommand& TriggerEventTest**

Description: This command generates a test event and sends it to the host.

Visibility:

**10.10.4.650 GenApi::IEnumerationT<TriggerModeEnums>& TriggerMode**

Description:

Controls whether or not trigger is active.

Visibility:

**10.10.4.651 GenApi::Integer& TriggerMultiplier**

Description: Specifies a multiplication factor for the incoming trigger pulses.

It is used generally used in conjunction with TriggerDivider to control the ratio of triggers that are accepted.

Visibility: Expert

**10.10.4.652 GenApi::IEnumerationT<TriggerOverlapEnums>& TriggerOverlap**

Description: Specifies the overlap mode of the trigger.

Visibility:

**10.10.4.653 GenApi::IEnumerationT<TriggerSelectorEnums>& TriggerSelector**

Description: Selects the type of trigger to configure.

Visibility:

**10.10.4.654 GenApi::ICommand& TriggerSoftware**

Description:

Generates an internal trigger if Trigger Source is set to Software.

Visibility:

**10.10.4.655 GenApi::IEnumerationT<TriggerSourceEnums>& TriggerSource**

Description:

Specifies the internal signal or physical input line to use as the trigger source.

Visibility:

**10.10.4.656 GenApi::IEnumerationT<UserOutputSelectorEnums>& UserOutputSelector**

Description: Selects which bit of the User Output register is set by UserOutputValue.

Visibility:

**10.10.4.657 GenApi::IBoolean& UserOutputValue**

Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).

Visibility:

**10.10.4.658 GenApi::IInteger& UserOutputValueAll**

Description: Returns the current status of all the user output status bits in a hexadecimal representation (UserOutput 0 status corresponds to bit 0, UserOutput 1 status with bit 1, etc).

This allows simultaneous reading of all user output statuses at once. Visibility:

**10.10.4.659 GenApi::IInteger& UserOutputValueAllMask**

Description: Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.

If the UserOutputValueAllMask feature is present, setting the user Output register using UserOutputValueAll will only change the bits that have a corresponding bit in the mask set to one. Visibility: Expert

**10.10.4.660 GenApi::IEnumerationT<UserSetDefaultEnums>& UserSetDefault**

Description:

Selects the feature User Set to load and make active by default when the device is restarted.

Visibility:

**10.10.4.661 GenApi::IBoolean& UserSetFeatureEnable**

Description: Whether or not the selected feature is saved to user sets.

Visibility:

**10.10.4.662 GenApi::ICommand& UserSetLoad**

Description:

Loads the User Set specified by UserSetSelector to the device and makes it active.

Visibility:

**10.10.4.663 GenApi::ICommand& UserSetSave**

Description:

Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.

Visibility:

**10.10.4.664 GenApi::IEnumerationT<UserSetSelectorEnums>& UserSetSelector**

Description:

Selects the feature User Set to load, save or configure.

Visibility:

**10.10.4.665 GenApi::IBoolean& V3\_3Enable**

Description: Internally generated 3.3V rail.

Enable to supply external circuits with power. This is different than standard logic outputs in that it is comparatively slow to switch but can supply a more significant amount of power. This is only available on some pins. Visibility:

#### 10.10.4.666 GenApi::IFloat& WhiteClip

Description: Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.

The video signal will never exceed the white clipping point: it will saturate at that level. Visibility: Expert

#### 10.10.4.667 GenApi::IEnumerationT<WhiteClipSelectorEnums>& WhiteClipSelector

Description: Selects which White Clip to control.

Visibility: Expert

#### 10.10.4.668 GenApi::Integer& Width

Description:

Width of the image provided by the device (in pixels).

Visibility:

#### 10.10.4.669 GenApi::Integer& WidthMax

Description:

Maximum width of the image (in pixels).

The dimension is calculated after horizontal binning. WidthMax does not take into account the current Region of interest (Width or OffsetX).

Visibility:

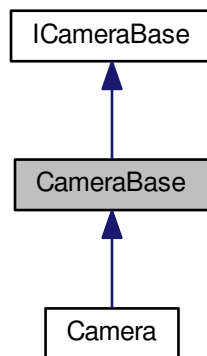
The documentation for this class was generated from the following file:

- [include/Camera.h](#)

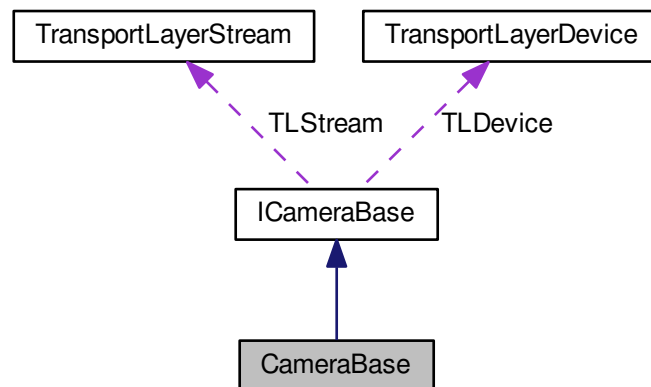
## 10.11 CameraBase Class Reference

The base class for the camera object.

Inheritance diagram for CameraBase:



Collaboration diagram for CameraBase:



### Public Member Functions

- virtual `~CameraBase` (void)  
*Virtual Destructor.*
- void `Init` ()  
*Connect to camera, retrieve XML and generate node map.*

- void [DelInit](#) ()  
*Disconnect camera port and free [GenICam](#) node map and GUI XML.*
- bool [IsInitialized](#) ()  
*Checks if camera is initialized.*
- bool [IsValid](#) ()  
*Checks a flag to determine if camera is still valid for use.*
- [GenApi::INodeMap](#) & [GetNodeMap](#) () const  
*Gets a reference to the node map that is generated from a [GenICam](#) XML file.*
- [GenApi::INodeMap](#) & [GetTLDeviceNodeMap](#) () const  
*Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Device module.*
- [GenApi::INodeMap](#) & [GetTLStreamNodeMap](#) () const  
*Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Stream module.*
- [GenApi::EAccessMode](#) [GetAccessMode](#) () const  
*Returns the access mode that the software has on the [Camera](#).*
- void [ReadPort](#) (uint64\_t iAddress, void \*pBuffer, size\_t iSize)
- void [WritePort](#) (uint64\_t iAddress, const void \*pBuffer, size\_t iSize)
- void [BeginAcquisition](#) ()  
*Starts the image acquisition engine.*
- void [EndAcquisition](#) ()  
*Stops the image acquisition engine.*
- [ImagePtr](#) [GetNextImage](#) (uint64\_t grabTimeout=[EVENT\\_TIMEOUT\\_INFINITE](#), uint64\_t streamID=0)  
*Gets the next image that was received by the transport layer.*
- [GenICam::gcstring](#) [GetUniqueID](#) ()  
*This returns a unique id string that identifies the camera.*
- bool [IsStreaming](#) () const  
*Returns true if the camera is currently streaming or false if it is not.*
- [GenICam::gcstring](#) [GetGuiXml](#) () const  
*Returns the GUI XML that can be passed into the [Spinnaker](#) GUI framework.*
- void [RegisterEvent](#) ([Event](#) &evtToRegister)  
*Registers a specific event for the camera.*
- void [RegisterEvent](#) ([Event](#) &evtToRegister, const [GenICam::gcstring](#) &eventName)  
*Registers a specific event for the camera.*
- void [UnregisterEvent](#) ([Event](#) &evtToUnregister)  
*Unregisters an event for the camera Events should be unregistered first before calling camera [DelInit\(\)](#).*
- unsigned int [GetNumImagesInUse](#) ()  
*Returns the number of images that are currently in use.*
- unsigned int [GetNumDataStreams](#) ()  
*Returns the number of streams that a device supports.*
- unsigned int [DiscoverMaxPacketSize](#) ()  
*Returns the largest packet size that can be safely used on the interface that device is connected to.*

## Protected Member Functions

- [CameraBase](#) (void)  
*Default constructor.*
- [CameraBase](#) (const [CameraBase](#) &)  
*Copy constructor.*
- [CameraBase](#) & [operator=](#) (const [CameraBase](#) &)  
*Assignment operator.*

## Friends

- class [InterfaceImpl](#)

## Additional Inherited Members

### 10.11.1 Detailed Description

The base class for the camera object.

### 10.11.2 Constructor & Destructor Documentation

#### 10.11.2.1 `virtual ~CameraBase ( void ) [virtual]`

Virtual Destructor.

#### 10.11.2.2 `CameraBase ( void ) [protected]`

Default constructor.

#### 10.11.2.3 `CameraBase ( const CameraBase & ) [protected]`

Copy constructor.

### 10.11.3 Member Function Documentation

#### 10.11.3.1 `void BeginAcquisition ( ) [virtual]`

Starts the image acquisition engine.

The camera must be initialized via a call to [Init\(\)](#) before starting an acquisition.

See also

[Init\(\)](#)

Implements [ICameraBase](#).

### 10.11.3.2 void DelInit ( ) [virtual]

Disconnect camera port and free [GenICam](#) node map and GUI XML.

Do not call more functions that access the remote device such as WritePort/ReadPort after calling [DelInit\(\)](#); Events should also be unregistered before calling camera [DelInit\(\)](#). Otherwise an exception will be thrown in the [DelInit\(\)](#) call and require the user to unregister events before the camera can be re-initialized again.

See also

[Init\(\)](#)  
[UnregisterEvent\(Event & evtToUnregister\)](#)

Implements [ICameraBase](#).

### 10.11.3.3 unsigned int DiscoverMaxPacketSize ( ) [virtual]

Returns the largest packet size that can be safely used on the interface that device is connected to.

Returns

The maximum packet size returned.

Implements [ICameraBase](#).

### 10.11.3.4 void EndAcquisition ( ) [virtual]

Stops the image acquisition engine.

If [EndAcquisition\(\)](#) is called without a prior call to [BeginAcquisition\(\)](#) an error message "Camera is not started" will be thrown. All Images that were acquired using [GetNextImage\(\)](#) need to be released first using `image->Release()` before calling [EndAcquisition\(\)](#). All buffers in the input pool and output queue will be discarded when [EndAcquisition\(\)](#) is called.

See also

[Init\(\)](#)  
[BeginAcquisition\(\)](#)  
[GetNextImage\( grabTimeout \)](#)  
[Image::Release\(\)](#)

Implements [ICameraBase](#).

### 10.11.3.5 GenApi::EAccessMode GetAccessMode ( ) const [virtual]

Returns the access mode that the software has on the [Camera](#).

The camera does not need to be initialized before calling this function.

See also

[Init\(\)](#)

Returns

An enumeration value indicating the access mode

Implements [ICameraBase](#).



### 10.11.3.6 `GenICam::gcstring GetGuiXml ( ) const` [virtual]

Returns the GUI XML that can be passed into the [Spinnaker](#) GUI framework.

#### Returns

[GenICam::gcstring](#) that represents the uncompressed GUI XML file

Implements [ICameraBase](#).

### 10.11.3.7 `ImagePtr GetNextImage ( uint64_t grabTimeout = EVENT_TIMEOUT_INFINITE, uint64_t streamID = 0 )` [virtual]

Gets the next image that was received by the transport layer.

This function will block indefinitely until an image arrives. Most cameras support one stream so the default streamID is 0 but if a camera supports multiple streams the user can input the streamID to select from which stream to grab images

#### See also

[Init\(\)](#)  
[BeginAcquisition\(\)](#)  
[EndAcquisition\(\)](#)

#### Parameters

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| <i>grabTimeout</i> | a 64bit value that represents a timeout in milliseconds |
| <i>streamID</i>    | The stream to grab the image.                           |

#### Returns

pointer to an [Image](#) object

Implements [ICameraBase](#).

### 10.11.3.8 `GenApi::INodeMap& GetNodeMap ( ) const` [virtual]

Gets a reference to the node map that is generated from a [GenICam](#) XML file.

The camera must be initialized by a call to [Init\(\)](#) first before a node map reference can be successfully acquired.

#### See also

[Init\(\)](#)

#### Returns

A reference to the [INodeMap](#).

Implements [ICameraBase](#).

#### 10.11.3.9 unsigned int GetNumDataStreams ( ) [virtual]

Returns the number of streams that a device supports.

##### Returns

The number of data streams

Implements [ICameraBase](#).

#### 10.11.3.10 unsigned int GetNumImagesInUse ( ) [virtual]

Returns the number of images that are currently in use.

Each of the images that are currently in use must be cleaned up with a call to `image->Release()` before calling `system->ReleaseInstance()`.

##### Returns

The number of images that needs to be cleaned up.

Implements [ICameraBase](#).

#### 10.11.3.11 GenApi::INodeMap& GetTLDeviceNodeMap ( ) const [virtual]

Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Device module.

The camera does not need to be initialized before acquiring this node map.

##### Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

#### 10.11.3.12 GenApi::INodeMap& GetTLStreamNodeMap ( ) const [virtual]

Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Stream module.

The camera does not need to be initialized before acquiring this node map.

##### Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

#### 10.11.3.13 `GenICam::gcstring GetUniqueID( )` [virtual]

This returns a unique id string that identifies the camera.

This is the camera serial number.

##### Returns

string that uniquely identifies the camera (serial number)

Implements [ICameraBase](#).

#### 10.11.3.14 `void Init( )` [virtual]

Connect to camera, retrieve XML and generate node map.

This function needs to be called before any camera related API calls such as [BeginAcquisition\(\)](#), [EndAcquisition\(\)](#), [GetNodeMap\(\)](#), [GetNextImage\(\)](#).

##### See also

[BeginAcquisition\(\)](#)  
[EndAcquisition\(\)](#)  
[GetNodeMap\(\)](#)  
[GetNextImage\(\)](#)

Implements [ICameraBase](#).

#### 10.11.3.15 `bool IsInitialized( )` [virtual]

Checks if camera is initialized.

This function needs to return true in order to retrieve a valid NodeMap from the [GetNodeMap\(\)](#) call.

##### See also

[GetNodeMap\(\)](#)

##### Returns

If camera is initialized or not

Implements [ICameraBase](#).

10.11.3.16 **bool IsStreaming ( ) const** [virtual]

Returns true if the camera is currently streaming or false if it is not.

See also

[Init\(\)](#)

Returns

returns true if camera is streaming and false otherwise.

Implements [ICameraBase](#).

10.11.3.17 **bool IsValid ( )** [virtual]

Checks a flag to determine if camera is still valid for use.

Returns

If camera is valid or not

Implements [ICameraBase](#).

10.11.3.18 **CameraBase& operator= ( const CameraBase & )** [protected]

Assignment operator.

10.11.3.19 **void ReadPort ( uint64\_t iAddress, void \* pBuffer, size\_t iSize )** [virtual]

Implements [ICameraBase](#).

10.11.3.20 **void RegisterEvent ( Event & evtToRegister )** [virtual]

Registers a specific event for the camera.

The camera has to be initialized first with a call to [Init\(\)](#) before registering for events.

See also

[Init\(\)](#)

Parameters

|                      |                                      |
|----------------------|--------------------------------------|
| <i>evtToRegister</i> | The event to register for the camera |
|----------------------|--------------------------------------|

Implements [ICameraBase](#).

10.11.3.21 `void RegisterEvent ( Event & evtToRegister, const GenICam::gcstring & eventName )` [virtual]

Registers a specific event for the camera.

See also

[Init\(\)](#)

Parameters

|                      |                                      |
|----------------------|--------------------------------------|
| <i>evtToRegister</i> | The event to register for the camera |
| <i>eventName</i>     | The event name to register           |

Implements [ICameraBase](#).

10.11.3.22 `void UnregisterEvent ( Event & evtToUnregister )` [virtual]

Unregisters an event for the camera Events should be unregistered first before calling camera [DeInit\(\)](#).

Otherwise an exception will be thrown in the [DeInit\(\)](#) call and require the user to unregister events before the camera can be re-initialized again.

See also

[DeInit\(\)](#)

Parameters

|                        |                                         |
|------------------------|-----------------------------------------|
| <i>evtToUnregister</i> | The event to unregister from the camera |
|------------------------|-----------------------------------------|

Implements [ICameraBase](#).

10.11.3.23 `void WritePort ( uint64_t iAddress, const void * pBuffer, size_t iSize )` [virtual]

Implements [ICameraBase](#).

## 10.11.4 Friends And Related Function Documentation

10.11.4.1 `friend class InterfacImpl` [friend]

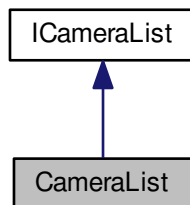
The documentation for this class was generated from the following file:

- [include/CameraBase.h](#)

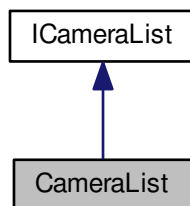
## 10.12 CameraList Class Reference

Used to hold a list of camera objects.

Inheritance diagram for CameraList:



Collaboration diagram for CameraList:



### Public Member Functions

- [CameraList](#) (void)  
*Default constructor.*
- virtual [~CameraList](#) (void)  
*Virtual destructor.*
- [CameraList](#) (const [CameraList](#) &iface)  
*Copy constructor.*
- [CameraList & operator=](#) (const [CameraList](#) &iface)  
*Assignment operator.*
- [CameraPtr operator\[\]](#) (unsigned int index)  
*Array subscription operators.*
- unsigned int [GetSize](#) () const  
*Returns the size of the camera list.*
- [CameraPtr GetByIndex](#) (unsigned int index) const

- Returns a pointer to a camera object at the "index".*
- [CameraPtr GetBySerial](#) (std::string serialNumber) const  
*Returns a pointer to a camera object with the specified serial number.*
- void [Clear](#) ()  
*Clears the list of cameras and destroys their corresponding reference counted objects.*
- void [RemoveByIndex](#) (unsigned int index)  
*Removes a camera at "index" and destroys its corresponding reference counted object.*
- void [RemoveBySerial](#) (std::string serialNumber)  
*Removes a camera using its serial number and destroys its corresponding reference counted object.*
- void [Append](#) ([CameraList](#) &otherList)  
*Appends a camera list to the current list.*

## Additional Inherited Members

### 10.12.1 Detailed Description

Used to hold a list of camera objects.

### 10.12.2 Constructor & Destructor Documentation

#### 10.12.2.1 [CameraList](#) ( void )

Default constructor.

#### 10.12.2.2 virtual [~CameraList](#) ( void ) [virtual]

Virtual destructor.

#### 10.12.2.3 [CameraList](#) ( const [CameraList](#) &iface )

Copy constructor.

### 10.12.3 Member Function Documentation

#### 10.12.3.1 void [Append](#) ( [CameraList](#) &otherList ) [virtual]

Appends a camera list to the current list.

##### Parameters

|                  |                                       |
|------------------|---------------------------------------|
| <i>otherList</i> | The other list to append to this list |
|------------------|---------------------------------------|

Implements [ICameraList](#).

### 10.12.3.2 void Clear ( ) [virtual]

Clears the list of cameras and destroys their corresponding reference counted objects.

This is necessary in order to clean up the parent interface. It is important that the camera list is destroyed or is cleared before calling `system->ReleaseInstance()` or else the call to `system->ReleaseInstance()` will result in an error message thrown that a reference to the camera is still held.

See also

[System:ReleaseInstance\(\)](#)

Implements [ICameraList](#).

### 10.12.3.3 CameraPtr GetByIndex ( unsigned int *index* ) const [virtual]

Returns a pointer to a camera object at the "index".

This function will throw a [Spinnaker](#) exception with `SPINNAKER_ERR_INVALID_PARAMETER` error if the input index is out of range.

Parameters

|              |                                                  |
|--------------|--------------------------------------------------|
| <i>index</i> | The index at which to retrieve the camera object |
|--------------|--------------------------------------------------|

Returns

A pointer to an camera object.

Implements [ICameraList](#).

### 10.12.3.4 CameraPtr GetBySerial ( std::string *serialNumber* ) const [virtual]

Returns a pointer to a camera object with the specified serial number.

This function will return a NULL [CameraPtr](#) if no matching camera serial is found.

Parameters

|                     |                                                    |
|---------------------|----------------------------------------------------|
| <i>serialNumber</i> | The serial number of the camera object to retrieve |
|---------------------|----------------------------------------------------|

Returns

A pointer to an camera object.

Implements [ICameraList](#).



10.12.3.5 `unsigned int GetSize ( ) const [virtual]`

Returns the size of the camera list.

The size is the number of [Camera](#) objects stored in the list.

#### Returns

An integer that represents the list size.

Implements [ICameraList](#).

10.12.3.6 `CameraList& operator= ( const CameraList & iface )`

Assignment operator.

10.12.3.7 `CameraPtr operator[] ( unsigned int index ) [virtual]`

Array subscription operators.

Implements [ICameraList](#).

10.12.3.8 `void RemoveByIndex ( unsigned int index ) [virtual]`

Removes a camera at "index" and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with `SPINNAKER_ERR_INVALID_PARAMETER` error if the input index is out of range.

#### Parameters

|              |                                                                |
|--------------|----------------------------------------------------------------|
| <i>index</i> | The index at which to remove the <a href="#">Camera</a> object |
|--------------|----------------------------------------------------------------|

Implements [ICameraList](#).

10.12.3.9 `void RemoveBySerial ( std::string serialNumber ) [virtual]`

Removes a camera using its serial number and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with `SPINNAKER_ERR_NOT_AVAILABLE` error if no matching camera serial is found.

#### Parameters

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| <i>serialNumber</i> | The serial number of the <a href="#">Camera</a> object to remove |
|---------------------|------------------------------------------------------------------|

Implements [ICameraList](#).

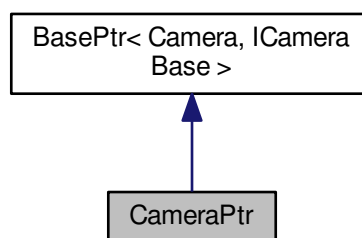
The documentation for this class was generated from the following file:

- include/[CameraList.h](#)

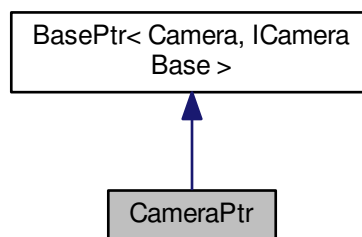
## 10.13 CameraPtr Class Reference

A reference tracked pointer to a camera object.

Inheritance diagram for CameraPtr:



Collaboration diagram for CameraPtr:



### Public Member Functions

- [CameraPtr](#) () throw ()  
*Default constructor.*
- [CameraPtr](#) (const int) throw ()  
*Default constructor.*
- virtual [~CameraPtr](#) (void)  
*Virtual destructor.*
- virtual [CameraPtr](#) & [operator=](#) (const int nMustBeNull)  
*Assignment operator.*

## Additional Inherited Members

### 10.13.1 Detailed Description

A reference tracked pointer to a camera object.

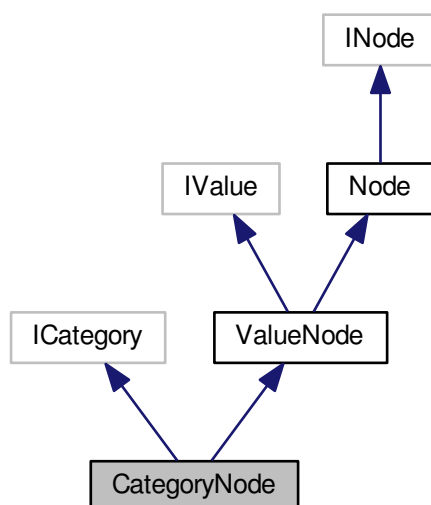
The documentation for this class was generated from the following file:

- [include/CameraPtr.h](#)

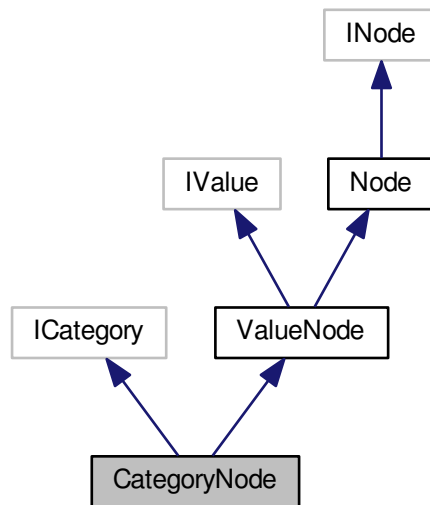
## 10.14 CategoryNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for CategoryNode:



Collaboration diagram for CategoryNode:



## Public Member Functions

- [CategoryNode](#) ( )
- [CategoryNode](#) (std::shared\_ptr< Node::NodeImpl > pCategory)
- virtual [~CategoryNode](#) ( )
- virtual void [GetFeatures](#) (FeatureList\_t &Features) const  
*Get all features of the category (including sub-categories)*
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.14.1 Detailed Description

[Interface](#) for string properties.

### 10.14.2 Constructor & Destructor Documentation

#### 10.14.2.1 [CategoryNode](#) ( )

#### 10.14.2.2 [CategoryNode](#) ( std::shared\_ptr< Node::NodeImpl > pCategory )

#### 10.14.2.3 virtual [~CategoryNode](#) ( ) [virtual]

### 10.14.3 Member Function Documentation

#### 10.14.3.1 virtual void [GetFeatures](#) ( FeatureList\_t & Features ) const [virtual]

Get all features of the category (including sub-categories)

10.14.3.2 `virtual void SetReference ( INode * pBase ) [virtual]`

overload SetReference for Value

Reimplemented from [ValueNode](#).

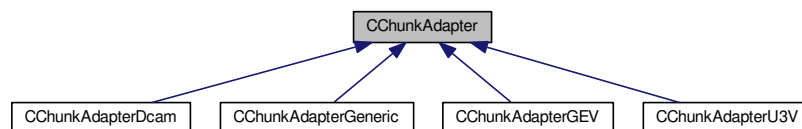
The documentation for this class was generated from the following file:

- include/SpinGenApi/[CategoryNode.h](#)

## 10.15 CChunkAdapter Class Reference

Connects a chunked buffer to a node map.

Inheritance diagram for CChunkAdapter:



### Public Member Functions

- `virtual ~CChunkAdapter ()`  
*Destructor.*
- `void AttachNodeMap (INodeMap *pNodeMap)`  
*Attaches to a node map and retrieves the chunk ports.*
- `void DetachNodeMap ()`  
*Detaches from the node map.*
- `virtual bool CheckBufferLayout (uint8_t *pBuffer, int64_t BufferLength)=0`  
*Checks if a buffer contains chunks in a known format.*
- `virtual void AttachBuffer (uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL)=0`  
*Attaches a buffer to the matching ChunkPort.*
- `void DetachBuffer ()`  
*Detaches a buffer.*
- `void UpdateBuffer (uint8_t *pBaseAddress)`  
*Updates the base address of the buffer.*
- `void ClearCaches ()`  
*Clears the chunk caches.*

### Protected Member Functions

- `CChunkAdapter (INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)`  
*Serves as default constructor.*

## Protected Attributes

- void \* [m\\_pChunkAdapter](#)

### 10.15.1 Detailed Description

Connects a chunked buffer to a node map.

### 10.15.2 Constructor & Destructor Documentation

10.15.2.1 `virtual ~CChunkAdapter ( )` `[virtual]`

Destructor.

10.15.2.2 `CChunkAdapter ( INodeMap * pNodeMap = NULL, int64_t MaxChunkCacheSize = -1 )` `[protected]`

Serves as default constructor.

### 10.15.3 Member Function Documentation

10.15.3.1 `virtual void AttachBuffer ( uint8_t * pBuffer, int64_t BufferLength, AttachStatistics_t * pAttachStatistics = NULL )` `[pure virtual]`

Attaches a buffer to the matching ChunkPort.

Implemented in [CChunkAdapterDcam](#), [CChunkAdapterGeneric](#), [CChunkAdapterGEV](#), and [CChunkAdapterU3V](#).

10.15.3.2 `void AttachNodeMap ( INodeMap * pNodeMap )`

Attaches to a node map and retrieves the chunk ports.

10.15.3.3 `virtual bool CheckBufferLayout ( uint8_t * pBuffer, int64_t BufferLength )` `[pure virtual]`

Checks if a buffer contains chunks in a known format.

Implemented in [CChunkAdapterDcam](#), [CChunkAdapterGeneric](#), [CChunkAdapterGEV](#), and [CChunkAdapterU3V](#).

10.15.3.4 `void ClearCaches ( )`

Clears the chunk caches.

#### 10.15.3.5 void DetachBuffer ( )

Detaches a buffer.

#### 10.15.3.6 void DetachNodeMap ( )

Detaches from the node map.

#### 10.15.3.7 void UpdateBuffer ( uint8\_t \* *pBaseAddress* )

Updates the base address of the buffer.

### 10.15.4 Member Data Documentation

#### 10.15.4.1 void\* m\_pChunkAdapter [protected]

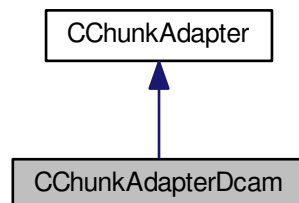
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapter.h](#)

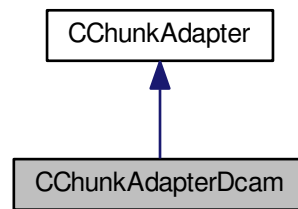
## 10.16 CChunkAdapterDcam Class Reference

Connects a chunked DCAM buffer to a node map.

Inheritance diagram for CChunkAdapterDcam:



Collaboration diagram for CChunkAdapterDcam:



## Public Member Functions

- [CChunkAdapterDcam](#) ([INodeMap](#) \*pNodeMap=NULL, [int64\\_t](#) MaxChunkCacheSize=-1)  
*Constructor.*
- virtual [~CChunkAdapterDcam](#) ()  
*Destructor.*
- virtual bool [CheckBufferLayout](#) ([uint8\\_t](#) \*pBuffer, [int64\\_t](#) BufferLength)  
*Checks if a buffer contains chunks in a known format.*
- virtual void [AttachBuffer](#) ([uint8\\_t](#) \*pBuffer, [int64\\_t](#) BufferLength, [AttachStatistics\\_t](#) \*pAttachStatistics=NULL)  
*Attaches a buffer to the matching ChunkPort.*
- bool [HasCRC](#) ([uint8\\_t](#) \*pBuffer, [int64\\_t](#) BufferLength)  
*Checks if buffer has a CRC attached.*
- bool [CheckCRC](#) ([uint8\\_t](#) \*pBuffer, [int64\\_t](#) BufferLength)  
*Checks CRC sum of buffer.*

## Additional Inherited Members

### 10.16.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

### 10.16.2 Constructor & Destructor Documentation

#### 10.16.2.1 [CChunkAdapterDcam](#) ( [INodeMap](#) \* *pNodeMap* = NULL, [int64\\_t](#) *MaxChunkCacheSize* = -1 )

Constructor.

#### 10.16.2.2 virtual [~CChunkAdapterDcam](#) ( ) [virtual]

Destructor.



### 10.16.3 Member Function Documentation

10.16.3.1 `virtual void AttachBuffer ( uint8_t * pBuffer, int64_t BufferLength, AttachStatistics_t * pAttachStatistics = NULL ) [virtual]`

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

10.16.3.2 `virtual bool CheckBufferLayout ( uint8_t * pBuffer, int64_t BufferLength ) [virtual]`

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

10.16.3.3 `bool CheckCRC ( uint8_t * pBuffer, int64_t BufferLength )`

Checks CRC sum of buffer.

10.16.3.4 `bool HasCRC ( uint8_t * pBuffer, int64_t BufferLength )`

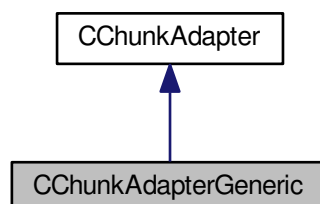
Checks if buffer has a CRC attached.

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

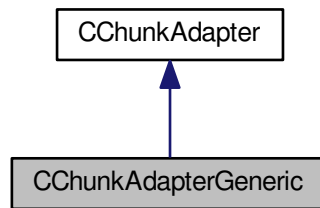
- `include/SpinGenApi/ChunkAdapterDcam.h`

## 10.17 CChunkAdapterGeneric Class Reference

Inheritance diagram for CChunkAdapterGeneric:



Collaboration diagram for CChunkAdapterGeneric:



## Public Member Functions

- [CChunkAdapterGeneric](#) ([INodeMap](#) \*pNodeMap=NULL, [int64\\_t](#) MaxChunkCacheSize=-1)
- virtual [~CChunkAdapterGeneric](#) ()
- virtual bool [CheckBufferLayout](#) ([uint8\\_t](#) \*pBuffer, [int64\\_t](#) BufferLength)  
*Checks if a buffer contains chunks in a known format.*
- virtual void [AttachBuffer](#) ([uint8\\_t](#) \*pBuffer, [int64\\_t](#) BufferLength, [AttachStatistics\\_t](#) \*pAttachStatistics=NULL)  
*Attaches a buffer to the matching ChunkPort.*
- virtual void [AttachBuffer](#) ([uint8\\_t](#) \*pBuffer, [SingleChunkData\\_t](#) \*ChunkData, [int64\\_t](#) NumChunks, [AttachStatistics\\_t](#) \*pAttachStatistics=NULL)
- virtual void [AttachBuffer](#) ([uint8\\_t](#) \*pBuffer, [SingleChunkDataStr\\_t](#) \*ChunkData, [int64\\_t](#) NumChunks, [AttachStatistics\\_t](#) \*pAttachStatistics=NULL)

## Additional Inherited Members

### 10.17.1 Constructor & Destructor Documentation

10.17.1.1 [CChunkAdapterGeneric](#) ( [INodeMap](#) \* pNodeMap = NULL, [int64\\_t](#) MaxChunkCacheSize = -1 )

10.17.1.2 virtual [~CChunkAdapterGeneric](#) ( ) [virtual]

### 10.17.2 Member Function Documentation

10.17.2.1 virtual void [AttachBuffer](#) ( [uint8\\_t](#) \* pBuffer, [int64\\_t](#) BufferLength, [AttachStatistics\\_t](#) \* pAttachStatistics = NULL ) [virtual]

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

10.17.2.2 `virtual void AttachBuffer ( uint8_t * pBuffer, SingleChunkData_t * ChunkData, int64_t NumChunks, AttachStatistics_t * pAttachStatistics = NULL ) [virtual]`

10.17.2.3 `virtual void AttachBuffer ( uint8_t * pBuffer, SingleChunkDataStr_t * ChunkData, int64_t NumChunks, AttachStatistics_t * pAttachStatistics = NULL ) [virtual]`

10.17.2.4 `virtual bool CheckBufferLayout ( uint8_t * pBuffer, int64_t BufferLength ) [virtual]`

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

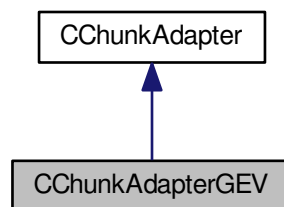
The documentation for this class was generated from the following file:

- `include/SpinGenApi/ChunkAdapterGeneric.h`

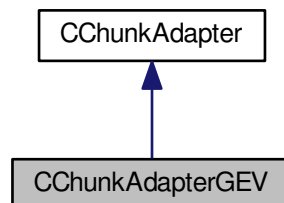
## 10.18 CChunkAdapterGEV Class Reference

Connects a chunked DCAM buffer to a node map.

Inheritance diagram for CChunkAdapterGEV:



Collaboration diagram for CChunkAdapterGEV:



## Public Member Functions

- [CChunkAdapterGEV](#) ([INodeMap](#) \*pNodeMap=NULL, int64\_t MaxChunkCacheSize=-1)  
*Constructor.*
- virtual [~CChunkAdapterGEV](#) ()  
*Destructor.*
- virtual bool [CheckBufferLayout](#) (uint8\_t \*pBuffer, int64\_t BufferLength)  
*Checks if a buffer contains chunks in a known format.*
- virtual void [AttachBuffer](#) (uint8\_t \*pBuffer, int64\_t BufferLength, [AttachStatistics\\_t](#) \*pAttachStatistics=NULL)  
*Attaches a buffer to the matching ChunkPort.*

## Additional Inherited Members

### 10.18.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

### 10.18.2 Constructor & Destructor Documentation

#### 10.18.2.1 [CChunkAdapterGEV](#) ( [INodeMap](#) \* *pNodeMap* = NULL, int64\_t *MaxChunkCacheSize* = -1 )

Constructor.

#### 10.18.2.2 virtual [~CChunkAdapterGEV](#) ( ) [virtual]

Destructor.

### 10.18.3 Member Function Documentation

#### 10.18.3.1 virtual void [AttachBuffer](#) ( uint8\_t \* *pBuffer*, int64\_t *BufferLength*, [AttachStatistics\\_t](#) \* *pAttachStatistics* = NULL ) [virtual]

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

#### 10.18.3.2 virtual bool [CheckBufferLayout](#) ( uint8\_t \* *pBuffer*, int64\_t *BufferLength* ) [virtual]

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

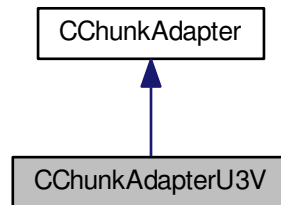
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGEV.h](#)

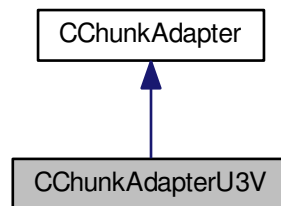
## 10.19 CChunkAdapterU3V Class Reference

Connects a chunked U3V buffer to a node map.

Inheritance diagram for CChunkAdapterU3V:



Collaboration diagram for CChunkAdapterU3V:



### Public Member Functions

- [CChunkAdapterU3V](#) ([INodeMap](#) \*pNodeMap=NULL, int64\_t MaxChunkCacheSize=-1)  
*Constructor.*
- virtual [~CChunkAdapterU3V](#) ()  
*Destructor.*
- virtual bool [CheckBufferLayout](#) (uint8\_t \*pBuffer, int64\_t BufferLength)  
*Checks if a buffer contains chunks in a known format.*
- virtual void [AttachBuffer](#) (uint8\_t \*pBuffer, int64\_t BufferLength, [AttachStatistics\\_t](#) \*pAttachStatistics=NULL)  
*Attaches a buffer to the matching ChunkPort.*

### Additional Inherited Members

#### 10.19.1 Detailed Description

Connects a chunked U3V buffer to a node map.

## 10.19.2 Constructor & Destructor Documentation

### 10.19.2.1 CChunkAdapterU3V ( INodeMap \* pNodeMap = NULL, int64\_t MaxChunkCacheSize = -1 )

Constructor.

### 10.19.2.2 virtual ~CChunkAdapterU3V ( ) [virtual]

Destructor.

## 10.19.3 Member Function Documentation

### 10.19.3.1 virtual void AttachBuffer ( uint8\_t \* pBuffer, int64\_t BufferLength, AttachStatistics\_t \* pAttachStatistics = NULL ) [virtual]

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

### 10.19.3.2 virtual bool CheckBufferLayout ( uint8\_t \* pBuffer, int64\_t BufferLength ) [virtual]

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

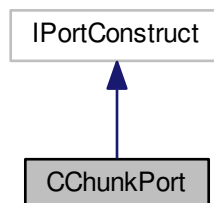
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

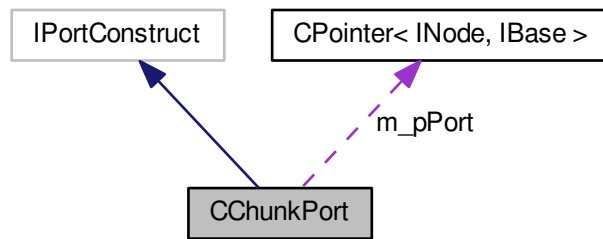
## 10.20 CChunkPort Class Reference

Port attachable to a chunk in a buffer.

Inheritance diagram for CChunkPort:



Collaboration diagram for CChunkPort:



## Public Member Functions

- [CChunkPort](#) ([IPort](#) \*pPort=NULL)  
*Constructor; can attach to a port.*
- [~CChunkPort](#) ()  
*Destructor; detaches from the port.*
- virtual [EAccessMode](#) [GetAccessMode](#) () const  
*Get the access mode of the node.*
- virtual [EInterfaceType](#) [GetPrincipallInterfaceType](#) () const  
*Get the type of the main interface of a node.*
- virtual void [Read](#) (void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Reads a chunk of bytes from the port.*
- virtual void [Write](#) (const void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Writes a chunk of bytes to the port.*
- virtual void [SetPortImpl](#) ([IPort](#) \*pPort)  
*Called from the port node to give the chunk port a pointer to itself.*
- virtual [EYesNo](#) [GetSwapEndianess](#) ()  
*Determines if the port adapter must perform an endianness swap.*
- void [InvalidateNode](#) ()
- bool [AttachPort](#) (::Spinnaker::GenApi::IPort \*pPort)  
*Attaches the ChunkPort to the Port.*
- void [DetachPort](#) ()  
*Detaches the ChunkPort to the Port.*
- void [AttachChunk](#) (uint8\_t \*pBaseAddress, int64\_t ChunkOffset, int64\_t [Length](#), bool Cache)  
*Attaches the Chunk to the ChunkPort.*
- void [DetachChunk](#) ()  
*Detaches the Chunk from the ChunkPort.*
- int [GetChunkIDLength](#) ()  
*Gets the ChunkID length.*
- bool [CheckChunkID](#) (uint8\_t \*pChunkIDBuffer, int ChunkIDLength)  
*Checks if a ChunkID matches.*
- bool [CheckChunkID](#) (uint64\_t ChunkID)  
*Checks if a ChunkID matches, version using uint64\_t ID representation.*
- void [UpdateBuffer](#) (uint8\_t \*pBaseAddress)  
*Updates the base address of the chunk.*
- void [ClearCache](#) ()  
*Clears the chunk cache.*

## Protected Attributes

- [CNodePtr m\\_pPort](#)
- `std::shared_ptr< PortAdapter > m_pPortAdapter`
- `void * m_pChunkPort`

### 10.20.1 Detailed Description

Port attachable to a chunk in a buffer.

### 10.20.2 Constructor & Destructor Documentation

#### 10.20.2.1 CChunkPort ( IPort \* *pPort* = NULL )

Constructor; can attach to a port.

#### 10.20.2.2 ~CChunkPort ( )

Destructor; detaches from the port.

### 10.20.3 Member Function Documentation

#### 10.20.3.1 void AttachChunk ( uint8\_t \* *pBaseAddress*, int64\_t *ChunkOffset*, int64\_t *Length*, bool *Cache* )

Attaches the Chunk to the ChunkPort.

#### 10.20.3.2 bool AttachPort ( ::Spinnaker::GenApi::IPort \* *pPort* )

Attaches the ChunkPort to the Port.

#### 10.20.3.3 bool CheckChunkID ( uint8\_t \* *pChunkIDBuffer*, int *ChunkIDLength* )

Checks if a ChunkID matches.

#### 10.20.3.4 bool CheckChunkID ( uint64\_t *ChunkID* )

Checks if a ChunkID matches, version using uint64\_t ID representation.

#### 10.20.3.5 void ClearCache ( )

Clears the chunk cache.



**10.20.3.6 void DetachChunk ( )**

Detaches the Chunk from the ChunkPort.

**10.20.3.7 void DetachPort ( )**

Detaches the ChunkPort to the Port.

**10.20.3.8 virtual EAccessMode GetAccessMode ( ) const** [virtual]

Get the access mode of the node.

**10.20.3.9 int GetChunkIDLength ( )**

Gets the ChunkID length.

**10.20.3.10 virtual EInterfaceType GetPrincipalInterfaceType ( ) const** [virtual]

Get the type of the main interface of a node.

**10.20.3.11 virtual EYesNo GetSwapEndianness ( )** [inline],[virtual]

Determines if the port adapter must perform an endianness swap.

**10.20.3.12 void InvalidateNode ( )****10.20.3.13 virtual void Read ( void \* *pBuffer*, int64\_t *Address*, int64\_t *Length* )** [virtual]

Reads a chunk of bytes from the port.

**10.20.3.14 virtual void SetPortImpl ( IPort \* *pPort* )** [virtual]

Called from the port node to give the chunk port a pointer to itself.

**10.20.3.15 void UpdateBuffer ( uint8\_t \* *pBaseAddress* )**

Updates the base address of the chunk.

**10.20.3.16 virtual void Write ( const void \* *pBuffer*, int64\_t *Address*, int64\_t *Length* )** [virtual]

Writes a chunk of bytes to the port.

## 10.20.4 Member Data Documentation

10.20.4.1 `void* m_pChunkPort` `[protected]`

10.20.4.2 `CNodePtr m_pPort` `[protected]`

10.20.4.3 `std::shared_ptr<PortAdapter> m_pPortAdapter` `[protected]`

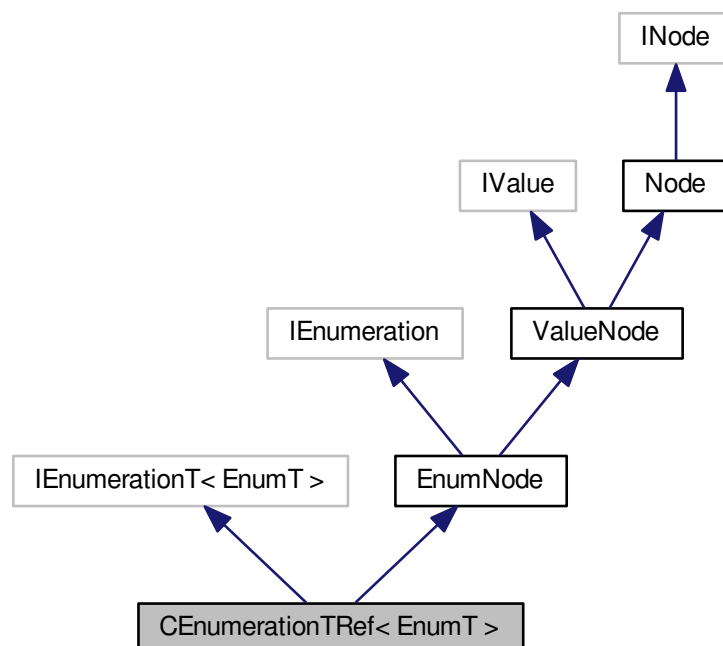
The documentation for this class was generated from the following file:

- `include/SpinGenApi/ChunkPort.h`

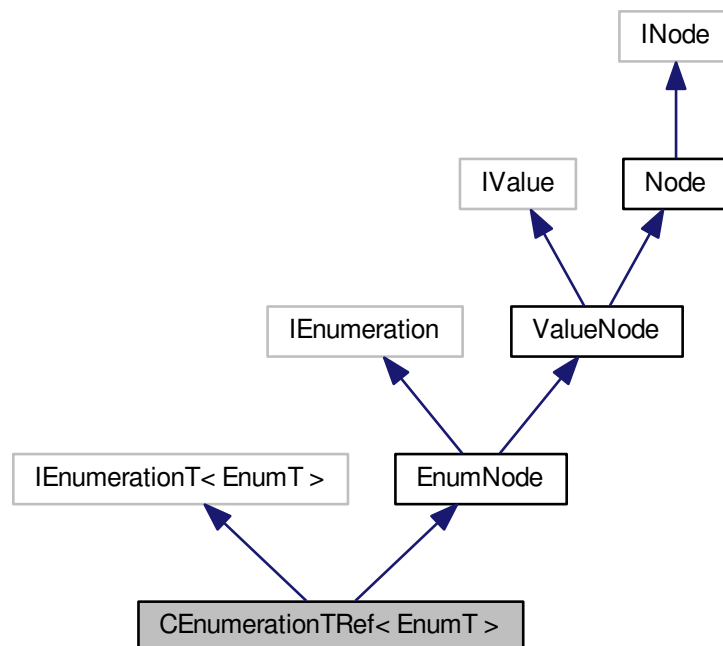
## 10.21 CEnumerationTRef< EnumT > Class Template Reference

[Interface](#) for string properties.

Inheritance diagram for CEnumerationTRef< EnumT >:



Collaboration diagram for CEnumerationTRef< EnumT >:



## Public Member Functions

- `CEnumerationTRef` ()
- `CEnumerationTRef` (std::shared\_ptr< Node::NodeImpl > pEnumeration)
- virtual `~CEnumerationTRef` ()
- virtual void `SetValue` (EnumT Value, bool `Verify`=true)  
Set node value.
- virtual `IEnumeration` & `operator=` (EnumT Value)  
Set node value.
- virtual EnumT `GetValue` (bool `Verify`=false, bool IgnoreCache=false)  
Get node value.
- virtual EnumT `operator()` ()  
Get node value.
- virtual `IEnumeration` & `operator=` (const `GenlCam::gcstring` &ValueStr)  
Set node value.
- virtual `IEnumEntry` \* `GetEntry` (const EnumT Value)  
returns the `EnumEntry` object belonging to the Value
- virtual `IEnumEntry` \* `GetEntry` (const int64\_t IntValue)  
Get an entry node by its IntValue.
- virtual `IEnumEntry` \* `GetCurrentEntry` (bool `Verify`=false, bool IgnoreCache=false)  
Get the current entry.
- virtual void `SetReference` (`INode` \*pBase)  
overload `SetReference` for `EnumerationT`

- virtual void [SetEnumReference](#) (int Index, [GenICam::gcstring](#) Name)  
*sets the Enum value corresponding to a value*
- virtual void [SetNumEnums](#) (int NumEnums)  
*sets the number of enum values*

## Additional Inherited Members

### 10.21.1 Detailed Description

```
template<class EnumT>
class Spinnaker::GenApi::CEnumerationTRef< EnumT >
```

[Interface](#) for string properties.

### 10.21.2 Constructor & Destructor Documentation

10.21.2.1 [CEnumerationTRef](#) ( )

10.21.2.2 [CEnumerationTRef](#) ( [std::shared\\_ptr](#)< [Node::NodeImpl](#) > *pEnumeration* )

10.21.2.3 virtual [~CEnumerationTRef](#) ( ) [virtual]

### 10.21.3 Member Function Documentation

10.21.3.1 virtual [IEnumEntry](#)\* [GetCurrentEntry](#) ( [bool](#) *Verify* = false, [bool](#) *IgnoreCache* = false ) [virtual]

Get the current entry.

Reimplemented from [EnumNode](#).

10.21.3.2 virtual [IEnumEntry](#)\* [GetEntry](#) ( [const](#) [EnumT](#) *Value* ) [virtual]

returns the EnumEntry object belonging to the Value

10.21.3.3 virtual [IEnumEntry](#)\* [GetEntry](#) ( [const](#) [int64\\_t](#) *IntValue* ) [virtual]

Get an entry node by its IntValue.

Reimplemented from [EnumNode](#).

10.21.3.4 virtual [EnumT](#) [GetValue](#) ( [bool](#) *Verify* = false, [bool](#) *IgnoreCache* = false ) [virtual]

Get node value.

## Parameters

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| <i>Verify</i>      | Enables Range verification (default = false). The AccessMode is always checked |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false)                |

## Returns

The value read

10.21.3.5 `virtual EnumT operator()( ) [virtual]`

Get node value.

10.21.3.6 `virtual IEnumeration& operator= ( EnumT Value ) [virtual]`

Set node value.

10.21.3.7 `virtual IEnumeration& operator= ( const GenICam::gcstring & ValueStr ) [virtual]`

Set node value.

Note : the operator= is not inherited thus the operator= versions from IEnumeration must be implemented again

Reimplemented from [EnumNode](#).

10.21.3.8 `virtual void SetEnumReference ( int Index, GenICam::gcstring Name ) [virtual]`

sets the Enum value corresponding to a value

10.21.3.9 `virtual void SetNumEnums ( int NumEnums ) [virtual]`

sets the number of enum values

10.21.3.10 `virtual void SetReference ( INode * pBase ) [virtual]`

overload SetReference for EnumerationT

Reimplemented from [EnumNode](#).

10.21.3.11 `virtual void SetValue ( EnumT Value, bool Verify=true ) [virtual]`

Set node value.

## Parameters

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>Value</i>  | The value to set                                           |
| <i>Verify</i> | Enables AccessMode and Range verification (default = true) |

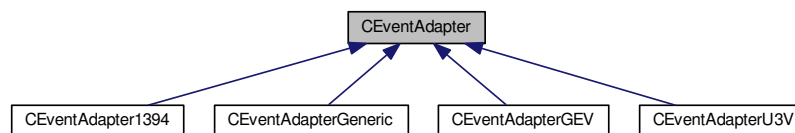
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumNodeT.h](#)

## 10.22 CEventAdapter Class Reference

Delivers Events to ports.

Inheritance diagram for CEventAdapter:



### Public Member Functions

- [CEventAdapter](#) ([INodeMap](#) \*pNodeMap=NULL)  
*Constructor.*
- virtual [~CEventAdapter](#) ()  
*Destructor.*
- virtual void [AttachNodeMap](#) ([INodeMap](#) \*pNodeMap)  
*Attaches to a node map and retrieves the chunk ports.*
- virtual void [DetachNodeMap](#) ()  
*Detaches from the node map.*
- virtual void [DeliverMessage](#) (const uint8\_t msg[], uint32\_t numBytes)=0  
*Deliver message.*

### Protected Attributes

- void \* [m\\_pEventAdapter](#)

#### 10.22.1 Detailed Description

Delivers Events to ports.

## 10.22.2 Constructor & Destructor Documentation

### 10.22.2.1 CEventAdapter ( INodeMap \* *pNodeMap* = NULL )

Constructor.

### 10.22.2.2 virtual ~CEventAdapter ( ) [virtual]

Destructor.

## 10.22.3 Member Function Documentation

### 10.22.3.1 virtual void AttachNodeMap ( INodeMap \* *pNodeMap* ) [virtual]

Attaches to a node map and retrieves the chunk ports.

### 10.22.3.2 virtual void DeliverMessage ( const uint8\_t *msg*[], uint32\_t *numBytes* ) [pure virtual]

Deliver message.

Implemented in [CEventAdapterGEV](#), [CEventAdapterU3V](#), [CEventAdapter1394](#), and [CEventAdapterGeneric](#).

### 10.22.3.3 virtual void DetachNodeMap ( ) [virtual]

Detaches from the node emap.

## 10.22.4 Member Data Documentation

### 10.22.4.1 void\* m\_pEventAdapter [protected]

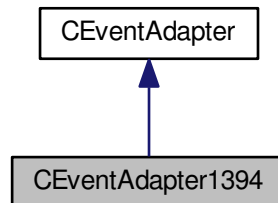
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter.h](#)

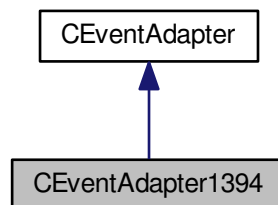
## 10.23 CEventAdapter1394 Class Reference

Distribute the events to the node map.

Inheritance diagram for CEventAdapter1394:



Collaboration diagram for CEventAdapter1394:



### Public Member Functions

- [CEventAdapter1394](#) ([INodeMap](#) \*pNodeMap=NULL)  
*constructor*
- virtual [~CEventAdapter1394](#) ()
- virtual void [DeliverMessage](#) (const uint8\_t msg[], uint32\_t numBytes)  
*Deliver message.*
- void [DeliverEventMessage](#) ([EventData1394](#) &[Event](#), uint32\_t numBytes)  
*distributes events to node map*

### Additional Inherited Members

#### 10.23.1 Detailed Description

Distribute the events to the node map.



### 10.23.2 Constructor & Destructor Documentation

10.23.2.1 **CEventAdapter1394** ( *INodeMap* \* *pNodeMap* = NULL ) [explicit]

constructor

10.23.2.2 **virtual** ~**CEventAdapter1394** ( ) [virtual]

### 10.23.3 Member Function Documentation

10.23.3.1 **void** DeliverEventMessage ( *EventData1394* & *Event*, *uint32\_t* *numBytes* )

distributes events to node map

10.23.3.2 **virtual void** DeliverMessage ( *const uint8\_t* *msg*[], *uint32\_t* *numBytes* ) [virtual]

Deliver message.

Implements [CEventAdapter](#).

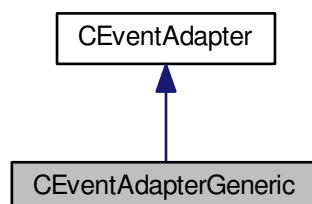
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter1394.h](#)

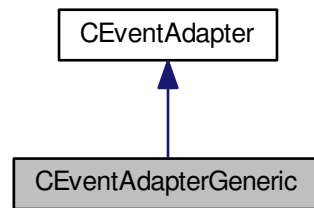
## 10.24 CEventAdapterGeneric Class Reference

Connects a generic event to a node map.

Inheritance diagram for CEventAdapterGeneric:



Collaboration diagram for CEventAdapterGeneric:



## Public Member Functions

- [CEventAdapterGeneric](#) ([INodeMap](#) \*pNodeMap=NULL)  
*Constructor.*
- virtual [~CEventAdapterGeneric](#) ()  
*Destructor.*
- virtual void [DeliverMessage](#) (const uint8\_t msg[], uint32\_t numBytes)  
*Deliver message.*
- virtual void [DeliverMessage](#) (const uint8\_t msg[], uint32\_t numBytes, const [GenICam::gcstring](#) &EventID)
- virtual void [DeliverMessage](#) (const uint8\_t msg[], uint32\_t numBytes, uint64\_t EventID)

## Additional Inherited Members

### 10.24.1 Detailed Description

Connects a generic event to a node map.

### 10.24.2 Constructor & Destructor Documentation

#### 10.24.2.1 [CEventAdapterGeneric](#) ( [INodeMap](#) \* *pNodeMap* = NULL )

Constructor.

#### 10.24.2.2 virtual [~CEventAdapterGeneric](#) ( ) [virtual]

Destructor.

### 10.24.3 Member Function Documentation

10.24.3.1 `virtual void DeliverMessage ( const uint8_t msg[], uint32_t numBytes )` `[virtual]`

Deliver message.

Implements [CEventAdapter](#).

10.24.3.2 `virtual void DeliverMessage ( const uint8_t msg[], uint32_t numBytes, const GenICam::gcstring & EventID )` `[virtual]`

10.24.3.3 `virtual void DeliverMessage ( const uint8_t msg[], uint32_t numBytes, uint64_t EventID )` `[virtual]`

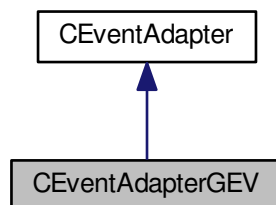
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGeneric.h](#)

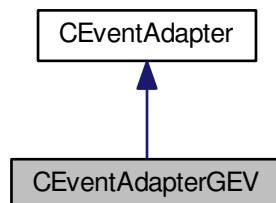
## 10.25 CEventAdapterGEV Class Reference

Connects a GigE [Event](#) to a node map.

Inheritance diagram for CEventAdapterGEV:



Collaboration diagram for CEventAdapterGEV:



## Public Member Functions

- [CEventAdapterGEV](#) ([INodeMap](#) \*pNodeMap=NULL)  
*Constructor.*
- virtual [~CEventAdapterGEV](#) ()  
*Destructor.*
- virtual void [DeliverMessage](#) (const uint8\_t msg[], uint32\_t numBytes)  
*Deliver message.*
- void [DeliverEventMessage](#) (const [GVCP\\_EVENT\\_REQUEST](#) \*pEvent)  
*Delivers the Events listed in the [Event](#) packet.*
- void [DeliverEventMessage](#) (const [GVCP\\_EVENTDATA\\_REQUEST](#) \*pEventData)  
*Delivers the [Event](#) + Data listed in the EventData packet.*

## Additional Inherited Members

### 10.25.1 Detailed Description

Connects a GigE [Event](#) to a node map.

### 10.25.2 Constructor & Destructor Documentation

#### 10.25.2.1 [CEventAdapterGEV](#) ( [INodeMap](#) \* *pNodeMap* = NULL )

Constructor.

#### 10.25.2.2 virtual [~CEventAdapterGEV](#) ( ) [virtual]

Destructor.

### 10.25.3 Member Function Documentation

#### 10.25.3.1 void [DeliverEventMessage](#) ( const [GVCP\\_EVENT\\_REQUEST](#) \* *pEvent* )

Delivers the Events listed in the [Event](#) packet.

#### 10.25.3.2 void [DeliverEventMessage](#) ( const [GVCP\\_EVENTDATA\\_REQUEST](#) \* *pEventData* )

Delivers the [Event](#) + Data listed in the EventData packet.

#### 10.25.3.3 virtual void [DeliverMessage](#) ( const uint8\_t *msg*[], uint32\_t *numBytes* ) [virtual]

Deliver message.

Implements [CEventAdapter](#).

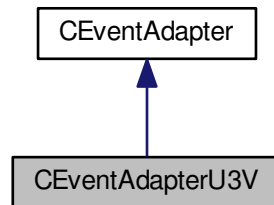
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

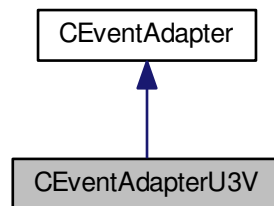
## 10.26 CEventAdapterU3V Class Reference

Connects a U3V [Event](#) to a node map.

Inheritance diagram for CEventAdapterU3V:



Collaboration diagram for CEventAdapterU3V:



### Public Member Functions

- [CEventAdapterU3V](#) ([INodeMap](#) \*pNodeMap=NULL)  
*Constructor.*
- virtual [~CEventAdapterU3V](#) ()  
*Destructor.*
- virtual void [DeliverMessage](#) (const uint8\_t msg[], uint32\_t numBytes)  
*Deliver message.*
- void [DeliverEventMessage](#) (const [U3V\\_EVENT\\_MESSAGE](#) \*pEventMessage)  
*Delivers the [Event](#) + Data listed in the packet.*

### Additional Inherited Members

#### 10.26.1 Detailed Description

Connects a U3V [Event](#) to a node map.

## 10.26.2 Constructor & Destructor Documentation

### 10.26.2.1 CEventAdapterU3V ( INodeMap \* *pNodeMap* = NULL )

Constructor.

### 10.26.2.2 virtual ~CEventAdapterU3V ( ) [virtual]

Destructor.

## 10.26.3 Member Function Documentation

### 10.26.3.1 void DeliverEventMessage ( const U3V\_EVENT\_MESSAGE \* *pEventMessage* )

Delivers the [Event](#) + Data listed in the packet.

### 10.26.3.2 virtual void DeliverMessage ( const uint8\_t *msg*[], uint32\_t *numBytes* ) [virtual]

Deliver message.

Implements [CEventAdapter](#).

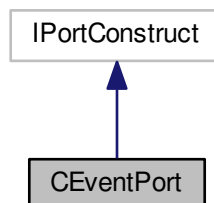
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

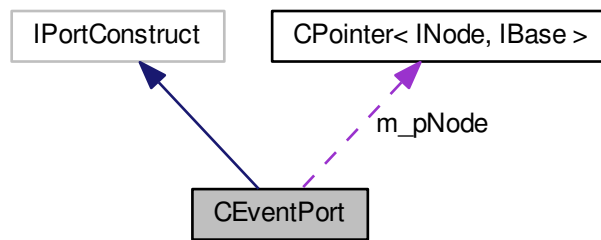
## 10.27 CEventPort Class Reference

Port attachable to an event.

Inheritance diagram for CEventPort:



Collaboration diagram for CEventPort:



## Public Member Functions

- [CEventPort](#) ([INode](#) \*pNode=NULL)  
*Constructor; can attach to a node.*
- [~CEventPort](#) ()  
*Destructor; detaches from the port.*
- virtual [EAccessMode](#) [GetAccessMode](#) () const  
*Get the access mode of the node.*
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const  
*Get the type of the main interface of a node.*
- virtual void [Read](#) (void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Reads a chunk of bytes from the port.*
- virtual void [Write](#) (const void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Writes a chunk of bytes to the port.*
- virtual void [SetPortImpl](#) (::Spinnaker::GenApi::IPort \*pPort)  
*Called from the port node to give the chunk port a pointer to itself.*
- virtual [EYesNo](#) [GetSwapEndianness](#) ()  
*Determines if the port adapter must perform an endianness swap.*
- void [InvalidateNode](#) ()
- bool [AttachNode](#) (::Spinnaker::GenApi::INode \*pNode)  
*Attaches to the [Node](#).*
- void [DetachNode](#) ()  
*Detaches from the [Node](#).*
- int [GetEventIDLength](#) ()  
*Gets the EventID length.*
- bool [CheckEventID](#) (uint8\_t \*pEventIDBuffer, int EventIDLength)  
*Checks if a EventID matches.*
- bool [CheckEventID](#) (uint64\_t EventID)  
*Checks if a EventID matches, version using uint64\_t ID representation.*
- void [AttachEvent](#) (uint8\_t \*pBaseAddress, int64\_t [Length](#))  
*Attaches the an [Event](#) to the EventPort.*
- void [DetachEvent](#) ()  
*Detaches the [Event](#) from the EventPort.*

## Protected Attributes

- [CNodePtr m\\_pNode](#)
- `std::shared_ptr< PortAdapter > m_pPortAdapter`
- `void * m_pEventPort`

### 10.27.1 Detailed Description

Port attachable to an event.

### 10.27.2 Constructor & Destructor Documentation

#### 10.27.2.1 `CEventPort ( INode * pNode = NULL )`

Constructor; can attach to a node.

#### 10.27.2.2 `~CEventPort ( )`

Destructor; detaches from the port.

### 10.27.3 Member Function Documentation

#### 10.27.3.1 `void AttachEvent ( uint8_t * pBaseAddress, int64_t Length )`

Attaches the an [Event](#) to the EventPort.

#### 10.27.3.2 `bool AttachNode ( ::Spinnaker::GenApi::INode * pNode )`

Attaches to the [Node](#).

#### 10.27.3.3 `bool CheckEventID ( uint8_t * pEventIDBuffer, int EventIDLength )`

Checks if a EventID matches.

#### 10.27.3.4 `bool CheckEventID ( uint64_t EventID )`

Checks if a EventID matches, version using uint64\_t ID representation.

#### 10.27.3.5 `void DetachEvent ( )`

Detaches the [Event](#) from the EventPort.



### 10.27.3.6 void DetachNode ( )

Detaches from the [Node](#).

### 10.27.3.7 virtual EAccessMode GetAccessMode ( ) const [virtual]

Get the access mode of the node.

### 10.27.3.8 int GetEventIDLength ( )

Gets the EventID length.

### 10.27.3.9 virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]

Get the type of the main interface of a node.

### 10.27.3.10 virtual EYesNo GetSwapEndianness ( ) [inline],[virtual]

Determines if the port adapter must perform an endianness swap.

### 10.27.3.11 void InvalidateNode ( )

### 10.27.3.12 virtual void Read ( void \* *pBuffer*, int64\_t *Address*, int64\_t *Length* ) [virtual]

Reads a chunk of bytes from the port.

### 10.27.3.13 virtual void SetPortImpl ( ::Spinnaker::GenApi::IPort \* *pPort* ) [virtual]

Called from the port node to give the chunk port a pointer to itself.

### 10.27.3.14 virtual void Write ( const void \* *pBuffer*, int64\_t *Address*, int64\_t *Length* ) [virtual]

Writes a chunk of bytes to the port.

## 10.27.4 Member Data Documentation

### 10.27.4.1 void\* m\_pEventPort [protected]

### 10.27.4.2 CNodePtr m\_pNode [protected]

### 10.27.4.3 std::shared\_ptr<PortAdapter> m\_pPortAdapter [protected]

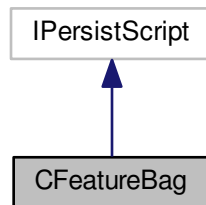
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventPort.h](#)

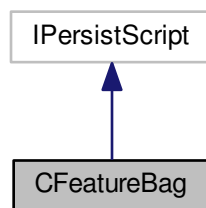
## 10.28 CFeatureBag Class Reference

Bag holding streamable features of a nodetree.

Inheritance diagram for CFeatureBag:



Collaboration diagram for CFeatureBag:



### Public Member Functions

- [CFeatureBag](#) ()
- virtual [~CFeatureBag](#) ()
- virtual void [SetInfo](#) ([GenICam::gcstring](#) &Info)  
*sets information about the node map*
- virtual void [PersistFeature](#) ([IValue](#) &item)  
*Stores a feature.*
- bool [LoadFromBag](#) ([INodeMap](#) \*pNodeMap, bool [Verify](#)=true, [GenICam::gcstring\\_vector](#) \*pErrorList=NULL)  
*Loads the features from the bag to the node tree.*
- int64\_t [StoreToBag](#) ([INodeMap](#) \*pNodeMap, const int MaxNumPersistSkriptEntries=-1)  
*Stores the streamable nodes to this feature bag.*
- bool [operator==](#) (const [CFeatureBag](#) &FeatureBag) const  
*compares the content of two feature bags*
- void \* [GetFeatureBagHandle](#) ()

### 10.28.1 Detailed Description

Bag holding streamable features of a nodetree.

### 10.28.2 Constructor & Destructor Documentation

10.28.2.1 **CFeatureBag** ( )

10.28.2.2 **virtual ~CFeatureBag** ( ) [virtual]

### 10.28.3 Member Function Documentation

10.28.3.1 **void\*** GetFeatureBagHandle ( )

10.28.3.2 **bool** LoadFromBag ( **INodeMap** \* *pNodeMap*, **bool** *Verify* = **true**, **GenICam::gcstring\_vector** \* *pErrorList* = **NULL** )

Loads the features from the bag to the node tree.

#### Parameters

|                   |                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------|
| <i>pNodeMap</i>   | The node map                                                                                        |
| <i>Verify</i>     | If true, all streamable features are read back                                                      |
| <i>pErrorList</i> | If an error occurs during loading the error message is stored in the list and the loading continues |

For *Verify*=true the list of names in the feature bag is replayed again. If a node is a selector it's value is set to the value from the feature bag. If not the value is read from the camera and compared with the value from the feature bag.

10.28.3.3 **bool** operator== ( **const CFeatureBag** & *FeatureBag* ) **const**

compares the content of two feature bags

10.28.3.4 **virtual void** PersistFeature ( **IValue** & *item* ) [virtual]

Stores a feature.

10.28.3.5 **virtual void** SetInfo ( **GenICam::gcstring** & *Info* ) [virtual]

sets information about the node map

10.28.3.6 **int64\_t** StoreToBag ( **INodeMap** \* *pNodeMap*, **const int** *MaxNumPersistSkriptEntries* = -1 )

Stores the streamable nodes to this feature bag.

## Parameters

|                                   |                                                                |
|-----------------------------------|----------------------------------------------------------------|
| <i>pNodeMap</i>                   | The node map to persist                                        |
| <i>MaxNumPersistSkriptEntries</i> | The max number of entries in the container; -1 means unlimited |

## Returns

number of entries in the bag

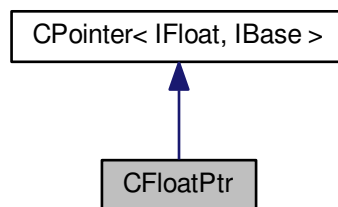
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Persistence.h](#)

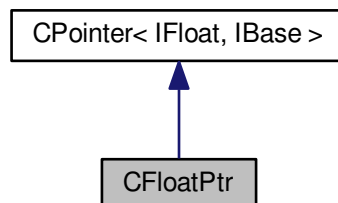
## 10.29 CFloatPtr Class Reference

SmartPointer for IFloat interface pointer.

Inheritance diagram for CFloatPtr:



Collaboration diagram for CFloatPtr:



## Public Member Functions

- [CFloatPtr](#) () throw ()  
*Default constructor.*
- [CFloatPtr](#) (IBase \*pB)  
*Constructor from IBase pointer type.*
- void [operator=](#) (IBase \*pB)  
*Assign IBase Pointer.*
- [IInteger](#) \* [GetIntAlias](#) ()  
*gets the interface of an integer alias node.*
- [IEnumeration](#) \* [GetEnumAlias](#) ()  
*gets the interface of an enum alias node.*

## Additional Inherited Members

### 10.29.1 Detailed Description

SmartPointer for IFloat interface pointer.

### 10.29.2 Constructor & Destructor Documentation

#### 10.29.2.1 [CFloatPtr](#) ( ) throw () [inline]

Default constructor.

#### 10.29.2.2 [CFloatPtr](#) ( IBase \* *pB* ) [inline]

Constructor from IBase pointer type.

### 10.29.3 Member Function Documentation

#### 10.29.3.1 [IEnumeration](#)\* [GetEnumAlias](#) ( ) [inline]

gets the interface of an enum alias node.

#### 10.29.3.2 [IInteger](#)\* [GetIntAlias](#) ( ) [inline]

gets the interface of an integer alias node.

#### 10.29.3.3 void [operator=](#) ( IBase \* *pB* ) [inline]

Assign IBase Pointer.

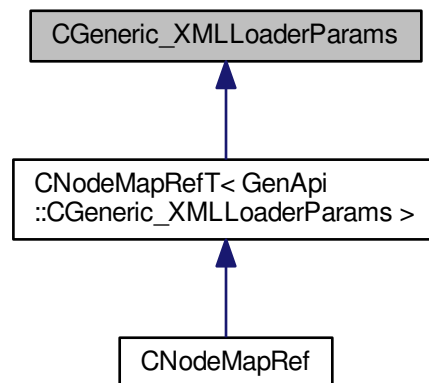
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Pointer.h](#)

## 10.30 CGeneric\_XMLLoaderParams Class Reference

Empty base class used by class [CNodeMapRef](#) as generic template argument.

Inheritance diagram for CGeneric\_XMLLoaderParams:



### Protected Member Functions

- virtual void [\\_Initialize](#) ([GenApi::INodeMap](#) \*)

### 10.30.1 Detailed Description

Empty base class used by class [CNodeMapRef](#) as generic template argument.

### 10.30.2 Member Function Documentation

10.30.2.1 virtual void [\\_Initialize](#) ( [GenApi::INodeMap](#) \* ) `[inline]`, `[protected]`, `[virtual]`

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

- include/SpinGenApi/[NodeMapRef.h](#)

## 10.31 CGlobalLock Class Reference

Named global lock which can be used over process boundaries.

## Public Member Functions

- [CGlobalLock](#) (const char \*pszName)  
*Creates a global lock object name pszName.*
- [CGlobalLock](#) (const [gcstring](#) &strName)  
*Creates a global lock object name strName.*
- [~CGlobalLock](#) ()
- bool [IsValid](#) (void) const  
*tests whether the lock is valid*
- bool [Lock](#) (unsigned int timeout\_ms)  
*enters the lock (may block)*
- bool [TryLock](#) (void)  
*tries to enter the lock and returns immediately when not possible*
- void [Unlock](#) (void)  
*leaves the lock*

## Protected Attributes

- long [m\\_DebugCount](#)

### 10.31.1 Detailed Description

Named global lock which can be used over process boundaries.

### 10.31.2 Constructor & Destructor Documentation

#### 10.31.2.1 [CGlobalLock](#) ( const char \* *pszName* ) [explicit]

Creates a global lock object name pszName.

In case an object with the same name already exists a reference to the existing object will be created. If pszName is NULL an unnamed object will be created.

#### 10.31.2.2 [CGlobalLock](#) ( const [gcstring](#) & *strName* ) [explicit]

Creates a global lock object name strName.

In case an object with the same name already exists a reference to the existing object will be created. If strName is empty an unnamed object will be created.

#### 10.31.2.3 [~CGlobalLock](#) ( )

### 10.31.3 Member Function Documentation

#### 10.31.3.1 bool [IsValid](#) ( void ) const

tests whether the lock is valid

### 10.31.3.2 `bool Lock ( unsigned int timeout_ms )`

enters the lock (may block)

### 10.31.3.3 `bool TryLock ( void )`

tries to enter the lock and returns immediately when not possible

### 10.31.3.4 `void Unlock ( void )`

leaves the lock

## 10.31.4 Member Data Documentation

### 10.31.4.1 `long m_DebugCount` `[mutable], [protected]`

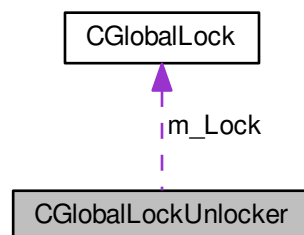
The documentation for this class was generated from the following file:

- `include/SpinGenApi/GCSynch.h`

## 10.32 CGlobalLockUnlocker Class Reference

Unlocks the global lock object on destruction.

Collaboration diagram for CGlobalLockUnlocker:



### Public Member Functions

- `CGlobalLockUnlocker (CGlobalLock &lock)`
- `~CGlobalLockUnlocker ()`
- `void UnlockEarly (void)`

*This function allows to unlock the object early before the object is destroyed.*



## Protected Attributes

- [CGlobalLock](#) & [m\\_Lock](#)
- bool [m\\_enabled](#)

### 10.32.1 Detailed Description

Unlocks the global lock object on destruction.

This is for automatic UNLOCKING only. We can't do automatic locking here since there is no returnvalue for constructors

### 10.32.2 Constructor & Destructor Documentation

10.32.2.1 `CGlobalLockUnlocker ( CGlobalLock & lock )` `[inline]`

10.32.2.2 `~CGlobalLockUnlocker ( )` `[inline]`

### 10.32.3 Member Function Documentation

10.32.3.1 `void UnlockEarly ( void )` `[inline]`

This function allows to unlock the object early before the object is destroyed.

### 10.32.4 Member Data Documentation

10.32.4.1 `bool m_enabled` `[protected]`

10.32.4.2 `CGlobalLock& m_Lock` `[protected]`

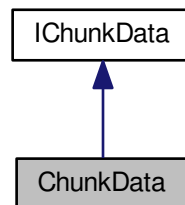
The documentation for this class was generated from the following file:

- `include/SpinGenApi/GCSynch.h`

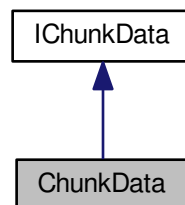
## 10.33 ChunkData Class Reference

The chunk data which contains additional information about an image.

Inheritance diagram for ChunkData:



Collaboration diagram for ChunkData:



### Public Member Functions

- [ChunkData](#) ()
- [ChunkData](#) (const [ChunkData](#) &src)
- virtual [~ChunkData](#) (void)
- void [SetChunks](#) ([GenApi::INodeMap](#) &pNodeMap)
- [float64\\_t](#) [GetBlackLevel](#) () const  
*Description: Returns the black level used to capture the image.*
- [int64\\_t](#) [GetFrameID](#) () const  
*Description: Returns the image frame ID.*
- [float64\\_t](#) [GetExposureTime](#) () const  
*Description: Returns the exposure time used to capture the image.*
- [int64\\_t](#) [GetTimestamp](#) () const  
*Description: Returns the Timestamp of the image.*
- [int64\\_t](#) [GetExposureEndLineStatusAll](#) () const

- Description: Returns the status of all the I/O lines at the end of exposure event.*

  - `int64_t GetWidth () const`

*Description: Returns the width of the image included in the payload.*
- `int64_t GetImage () const`

*Description: Returns the image payload.*
- `int64_t GetHeight () const`

*Description: Returns the height of the image included in the payload.*
- `float64_t GetGain () const`

*Description: Returns the gain used to capture the image.*
- `int64_t GetSequencerSetActive () const`

*Description: Returns the index of the active set of the running sequencer included in the payload.*
- `int64_t GetCRC () const`

*Description: Returns the CRC of the image payload.*
- `int64_t GetOffsetX () const`

*Description: Returns the Offset X of the image included in the payload.*
- `int64_t GetOffsetY () const`

*Description: Returns the Offset Y of the image included in the payload.*
- `int64_t GetSerialDataLength () const`

*Description: Returns the length of the received serial data that was included in the payload.*
- `int64_t GetPartSelector () const`

*Description: Selects the part to access in chunk data in a multipart transmission.*
- `int64_t GetPixelDynamicRangeMin () const`

*Description: Returns the minimum value of dynamic range of the image included in the payload.*
- `int64_t GetPixelDynamicRangeMax () const`

*Description: Returns the maximum value of dynamic range of the image included in the payload.*
- `int64_t GetTimestampLatchValue () const`

*Description: Returns the last Timestamp latched with the TimestampLatch command.*
- `int64_t GetLineStatusAll () const`

*Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.*
- `int64_t GetCounterValue () const`

*Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.*
- `float64_t GetTimerValue () const`

*Description: Returns the value of the selected Timer at the time of the FrameStart internal event.*
- `int64_t GetScanLineSelector () const`

*Description: Index for vector representation of one chunk value per line in an image.*
- `int64_t GetEncoderValue () const`

*Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.*
- `int64_t GetLinePitch () const`

*Description: Returns the LinePitch of the image included in the payload.*
- `int64_t GetTransferBlockID () const`

*Description: Returns the unique identifier of the transfer block used to transport the payload.*
- `int64_t GetTransferQueueCurrentBlockCount () const`

*Description: Returns the current number of blocks in the transfer queue.*
- `int64_t GetStreamChannelID () const`

*Description: Returns identifier of the stream channel used to carry the block.*
- `float64_t GetScan3dCoordinateScale () const`

*Description: Returns the Scale for the selected coordinate axis of the image included in the payload.*
- `float64_t GetScan3dCoordinateOffset () const`

*Description: Returns the Offset for the selected coordinate axis of the image included in the payload.*
- `float64_t GetScan3dInvalidDataValue () const`

*Description: Returns the Invalid Data Value used for the image included in the payload.*

- [float64\\_t GetScan3dAxisMin](#) () const

*Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.*

- [float64\\_t GetScan3dAxisMax](#) () const

*Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.*

- [float64\\_t GetScan3dTransformValue](#) () const

*Description: Returns the transform value.*

- [float64\\_t GetScan3dCoordinateReferenceValue](#) () const

*Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.*

- [int64\\_t GetInferenceResult](#) () const

*Description: Visibility: Expert.*

- [float64\\_t GetInferenceConfidence](#) () const

*Description: Visibility: Expert.*

## Additional Inherited Members

### 10.33.1 Detailed Description

The chunk data which contains additional information about an image.

### 10.33.2 Constructor & Destructor Documentation

#### 10.33.2.1 [ChunkData](#) ( )

#### 10.33.2.2 [ChunkData](#) ( const [ChunkData](#) & *src* )

#### 10.33.2.3 virtual [~ChunkData](#) ( void ) [virtual]

### 10.33.3 Member Function Documentation

#### 10.33.3.1 [float64\\_t GetBlackLevel](#) ( ) const [virtual]

Description: Returns the black level used to capture the image.

Visibility:

Implements [IChunkData](#).

#### 10.33.3.2 [int64\\_t GetCounterValue](#) ( ) const [virtual]

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

Implements [IChunkData](#).

#### 10.33.3.3 int64\_t GetCRC ( ) const [virtual]

Description: Returns the CRC of the image payload.

Visibility:

Implements [IChunkData](#).

#### 10.33.3.4 int64\_t GetEncoderValue ( ) const [virtual]

Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.

Visibility: Expert

Implements [IChunkData](#).

#### 10.33.3.5 int64\_t GetExposureEndLineStatusAll ( ) const [virtual]

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

Implements [IChunkData](#).

#### 10.33.3.6 float64\_t GetExposureTime ( ) const [virtual]

Description: Returns the exposure time used to capture the image.

Visibility:

Implements [IChunkData](#).

#### 10.33.3.7 int64\_t GetFrameID ( ) const [virtual]

Description: Returns the image frame ID.

Visibility:

Implements [IChunkData](#).

#### 10.33.3.8 float64\_t GetGain ( ) const [virtual]

Description: Returns the gain used to capture the image.

Visibility:

Implements [IChunkData](#).

10.33.3.9 `int64_t GetHeight ( ) const [virtual]`

Description: Returns the height of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.10 `int64_t GetImage ( ) const [virtual]`

Description: Returns the image payload.

Visibility:

Implements [IChunkData](#).

10.33.3.11 `float64_t GetInferenceConfidence ( ) const [virtual]`

Description: Visibility: Expert.

Implements [IChunkData](#).

10.33.3.12 `int64_t GetInferenceResult ( ) const [virtual]`

Description: Visibility: Expert.

Implements [IChunkData](#).

10.33.3.13 `int64_t GetLinePitch ( ) const [virtual]`

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.14 `int64_t GetLineStatusAll ( ) const [virtual]`

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.15 `int64_t GetOffsetX ( ) const [virtual]`

Description: Returns the Offset X of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.16 `int64_t GetOffsetY ( ) const [virtual]`

Description: Returns the Offset Y of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.17 `int64_t GetPartSelector ( ) const [virtual]`

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.18 `int64_t GetPixelDynamicRangeMax ( ) const [virtual]`

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.19 `int64_t GetPixelDynamicRangeMin ( ) const [virtual]`

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.20 `float64_t GetScan3dAxisMax ( ) const [virtual]`

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

**10.33.3.21** `float64_t GetScan3dAxisMin ( ) const [virtual]`

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

**10.33.3.22** `float64_t GetScan3dCoordinateOffset ( ) const [virtual]`

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

**10.33.3.23** `float64_t GetScan3dCoordinateReferenceValue ( ) const [virtual]`

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.

Visibility: Expert

Implements [IChunkData](#).

**10.33.3.24** `float64_t GetScan3dCoordinateScale ( ) const [virtual]`

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

**10.33.3.25** `float64_t GetScan3dInvalidDataValue ( ) const [virtual]`

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

**10.33.3.26** `float64_t GetScan3dTransformValue ( ) const [virtual]`

Description: Returns the transform value.

Visibility: Expert

Implements [IChunkData](#).



10.33.3.27 `int64_t GetScanLineSelector ( ) const [virtual]`

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.28 `int64_t GetSequencerSetActive ( ) const [virtual]`

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.29 `int64_t GetSerialDataLength ( ) const [virtual]`

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.30 `int64_t GetStreamChannelID ( ) const [virtual]`

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.31 `float64_t GetTimerValue ( ) const [virtual]`

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.32 `int64_t GetTimestamp ( ) const [virtual]`

Description: Returns the Timestamp of the image.

Visibility:

Implements [IChunkData](#).

10.33.3.33 `int64_t GetTimestampLatchValue ( ) const [virtual]`

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.34 `int64_t GetTransferBlockID ( ) const [virtual]`

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.35 `int64_t GetTransferQueueCurrentBlockCount ( ) const [virtual]`

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.36 `int64_t GetWidth ( ) const [virtual]`

Description: Returns the width of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.37 `void SetChunks ( GenApi::INodeMap & pNodeMap ) [virtual]`

Implements [IChunkData](#).

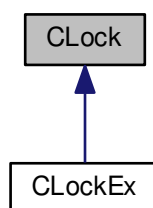
The documentation for this class was generated from the following file:

- [include/ChunkData.h](#)

## 10.34 CLock Class Reference

A lock class.

Inheritance diagram for CLock:



### Public Member Functions

- [CLock](#) ()  
*Constructor.*
- [CLock](#) (void \*pLock)  
*Constructor.*
- [~CLock](#) ()  
*Destructor.*
- bool [TryLock](#) ()  
*tries to enter the critical section; returns true if successful*
- void [Lock](#) ()  
*enters the critical section (may block)*
- void [Unlock](#) ()  
*leaves the critical section*

### Protected Attributes

- void \* [m\\_lock](#)
- bool [m\\_bOwnLock](#)

### Friends

- class [NodeMap](#)

#### 10.34.1 Detailed Description

A lock class.

## 10.34.2 Constructor & Destructor Documentation

### 10.34.2.1 CLock ( )

Constructor.

### 10.34.2.2 CLock ( void \* *pLock* )

Constructor.

### 10.34.2.3 ~CLock ( )

Destructor.

## 10.34.3 Member Function Documentation

### 10.34.3.1 void Lock ( )

enters the critical section (may block)

### 10.34.3.2 bool TryLock ( )

tries to enter the critical section; returns true if successful

### 10.34.3.3 void Unlock ( )

leaves the critical section

## 10.34.4 Friends And Related Function Documentation

### 10.34.4.1 friend class NodeMap [friend]

## 10.34.5 Member Data Documentation

### 10.34.5.1 bool m\_bOwnLock [protected]

### 10.34.5.2 void\* m\_lock [protected]

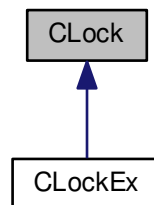
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

## 10.35 CLock Class Reference

A lock class.

Inheritance diagram for CLock:



### Public Member Functions

- [CLock](#) ()  
*Constructor.*
- [~CLock](#) ()  
*Destructor.*
- bool [TryLock](#) ()  
*tries to enter the critical section; returns true if successful*
- void [Lock](#) ()  
*enters the critical section (may block)*
- void [Unlock](#) ()  
*leaves the critical section*

### 10.35.1 Detailed Description

A lock class.

### 10.35.2 Constructor & Destructor Documentation

#### 10.35.2.1 CLock ( )

Constructor.

#### 10.35.2.2 ~CLock ( )

Destructor.

### 10.35.3 Member Function Documentation

#### 10.35.3.1 void Lock ( )

enters the critical section (may block)

#### 10.35.3.2 bool TryLock ( )

tries to enter the critical section; returns true if successful

#### 10.35.3.3 void Unlock ( )

leaves the critical section

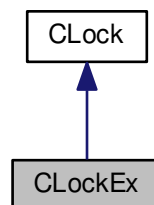
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

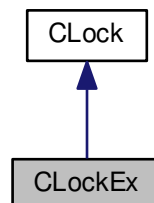
## 10.36 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



### Protected Attributes

- void \* [m\\_lockEx](#)

### Additional Inherited Members

#### 10.36.1 Detailed Description

This class is for testing purposes only.

It should not be used for client code because it exists only for Windows but not for Linux since it uses internal data structures of a Win32 object

#### 10.36.2 Member Data Documentation

10.36.2.1 void\* [m\\_lockEx](#) `[protected]`

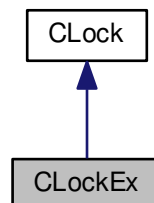
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

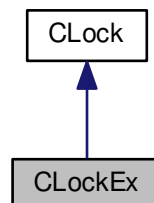
## 10.37 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



## Additional Inherited Members

### 10.37.1 Detailed Description

This class is for testing purposes only.

It should not be used for client code because it exists only for Windows but not for Linux since it uses internal data structures of a Win32 object

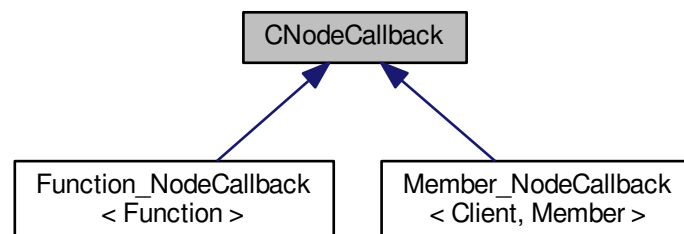
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

## 10.38 CNodeCallback Class Reference

callback body instance for INode pointers

Inheritance diagram for CNodeCallback:



## Public Member Functions

- [CNodeCallback](#) ([INode](#) \*pNode, [ECallbackType](#) CallbackType)
- virtual [~CNodeCallback](#) ()  
*virtual destructor*
- virtual void [operator\(\)](#) ([ECallbackType](#) CallbackType) const =0  
*fires the callback if th type is right*
- virtual void [Destroy](#) ()=0  
*destroys the object*
- [INode](#) \* [GetNode](#) ()  
*returns the node the callback is registered to*
- [ECallbackType](#) [GetCallbackType](#) ()



## Protected Attributes

- [INode \\* m\\_pNode](#)  
*the node were the callback is installed*
- [ECallbackType m\\_CallbackType](#)  
*the type of the callback*

### 10.38.1 Detailed Description

callback body instance for INode pointers

### 10.38.2 Constructor & Destructor Documentation

10.38.2.1 **CNodeCallback** ( [INode \\* pNode](#), [ECallbackType CallbackType](#) ) [inline]

10.38.2.2 **virtual ~CNodeCallback** ( ) [inline],[virtual]

virtual destructor

### 10.38.3 Member Function Documentation

10.38.3.1 **virtual void Destroy** ( ) [pure virtual]

destroys the object

Implemented in [Member\\_NodeCallback< Client, Member >](#), and [Function\\_NodeCallback< Function >](#).

10.38.3.2 **ECallbackType GetCallbackType** ( ) [inline]

10.38.3.3 **INode\* GetNode** ( ) [inline]

returns the node the callback is registered to

10.38.3.4 **virtual void operator()** ( [ECallbackType CallbackType](#) ) **const** [pure virtual]

fires the callback if th type is right

Implemented in [Member\\_NodeCallback< Client, Member >](#), and [Function\\_NodeCallback< Function >](#).

### 10.38.4 Member Data Documentation

10.38.4.1 **ECallbackType m\_CallbackType** [protected]

the type of the callback

#### 10.38.4.2 `Inode* m_pNode` [protected]

the node were the callback is installed

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

- include/SpinGenApi/[NodeCallback.h](#)

## 10.39 CNodeMapFactory Class Reference

The node map factory is used for creating node maps from camera description files.

### Classes

- struct [NodeStatistics\\_t](#)

### Public Member Functions

- [CNodeMapFactory](#) ()  
*Creates an empty node map factory for assigning a non-empty node map factory later.*
- virtual [~CNodeMapFactory](#) ()  
*Destroys the node map factory data if all references to the data have been released.*
- [CNodeMapFactory](#) (const [CNodeMapFactory](#) &)  
*Creates another reference to the node map factory data.*
- [CNodeMapFactory](#) & operator= (const [CNodeMapFactory](#) &)  
*Creates another reference to the assigned node map factory data.*
- [CNodeMapFactory](#) (EContentType\_t FileType, const [GenlCam::gcstring](#) &FileName, [ECacheUsage\\_t](#) CacheUsage=[CacheUsage\\_Automatic](#), bool SuppressStringsOnLoad=false)  
*Creates the node map factory and simply stores the full path to the provided camera description file data.*
- [CNodeMapFactory](#) (EContentType\_t ContentType, const void \*pData, size\_t DataSize, [ECacheUsage\\_t](#) CacheUsage=[CacheUsage\\_Automatic](#), bool SuppressStringsOnLoad=false)  
*Creates the node map factory and simply stores the pointer and the size of the provided camera description file data.*
- [CNodeMapFactory](#) (const [GenlCam::gcstring](#) &XmlData, [ECacheUsage\\_t](#) CacheUsage=[CacheUsage\\_Automatic](#), bool SuppressStringsOnLoad=false)  
*Creates the node map factory and copies the provided camera description file string.*
- bool [IsEmpty](#) () const  
*Returns true if nothing is loaded ([IsLoaded\(\)](#)) and no source data is available, e.g.*
- void [AddInjectionData](#) ([CNodeMapFactory](#) &injectionData)  
*Adds a node map factory representing a camera description file to inject.*
- void [LoadAndInject](#) ()  
*Advanced: Loads, Parses, and Injects the camera description files recursively.*
- bool [IsLoaded](#) () const  
*Can be used to check whether the [LoadAndInject\(\)](#) processing step has been performed.*
- [CNodeMapFactory](#) [ExtractSubtree](#) (const [GenlCam::gcstring](#) &SubTreeRootNodeName, bool doRename=[ToRoot=false](#))  
*The name of the node that represents the root of the subtree that shall be extracted.*
- void [Preprocess](#) ()

*Advanced: Creates the preprocessed memory internal representation of the camera description file(s), the CNode↔DataMap (not part of the public interface).*

- bool [IsPreprocessed](#) () const  
*Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.*
- void [ReleaseCameraDescriptionFileData](#) ()  
*Advanced: Releases any in constructors provided camera description file data buffers or files.*
- bool [IsCameraDescriptionFileDataReleased](#) () const  
*Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.*
- [INodeMap](#) \* [CreateNodeMap](#) (const [GenlCam::gcstring](#) &DeviceName="Device", bool DoReleaseCamera↔DescriptionFileData=true)  
*Creates a node map from the preprocessed memory internal representation of the camera description file(s).*
- [INodeMap](#) \* [CreateNodeMap](#) (CLOCK &UserProvidedLock, const [GenlCam::gcstring](#) &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true)  
*Creates a node map from the preprocessed memory internal representation of the camera description file(s).*
- void [GetSupportedSchemaVersions](#) ([GenlCam::gcstring\\_vector](#) &SchemaVersions) const
- [GenlCam::gcstring](#) [ToString](#) () const  
*Outputs the pre-processed node map in string form (for debug purpose)*
- [GenlCam::gcstring](#) [ToXml](#) () const  
*Outputs the pre-processed node map in XML form (mainly for debug purpose)*
- void [GetNodeStatistics](#) ([NodeStatistics\\_t](#) &NodeStatistics)
- const [GenlCam::gcstring](#) [ApplyStyleSheet](#) (const [GenlCam::gcstring](#) &StyleSheetFileName)  
*Applies a style sheet to the pre-processed node map.*

## Static Public Member Functions

- static [INodeMap](#) \* [CreateEmptyNodeMap](#) ()  
*Creates an empty node map usable as placeholder, e.g.*
- static bool [ClearCache](#) ()  
*Deletes all preprocessed camera description files from the cache.*
- static [CNodeDataMap](#) \* [CreateNodeDataFromNodeMap](#) ([INodeMap](#) \*pNodeMap)

### 10.39.1 Detailed Description

The node map factory is used for creating node maps from camera description files.

#### Examples

```
// Simple node map creation from buffer, downloaded from a device for instance.
CNodeMapFactory cameraNodeMapFactory( ContentType_ZippedXml, buffer,
bufferSize);

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).

// Node map creation with injecting additional xml fragments.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename1));
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename2));

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).
```

```

// Node map creation and additional extraction of a category subtree.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
// Extract a subtree for later chunk parsing.
CNodeMapFactory chunkDataNodeMapFactory = cameraParameters.ExtractSubtree("ChunkData");

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraParameters.CreateNodeMap();
// The next step is attaching the device port (not shown).

// Node map creation with injecting additional xml fragments and additional extraction of a category
// subtree.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename1));
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename2));
CNodeMapFactory chunkDataNodeMapFactory = cameraNodeMapFactory.ExtractSubtree("ChunkData");

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).

// A node map factory can create multiple node maps from the provided camera description file(s).
for(int i = 0; i < 20; ++i)
{
    INodeMap* pNodeMapChunks = chunkDataNodeMapFactory.CreateNodeMap();
    //...
}

```

#### Attention

The is [CNodeMapFactory](#) not thread-safe.

You need to take care when camera description file data can be actually be freed, see method documentation of the node map factory for more detail.

## 10.39.2 Constructor & Destructor Documentation

### 10.39.2.1 CNodeMapFactory ( )

Creates an empty node map factory for assigning a non-empty node map factory later.

### 10.39.2.2 virtual ~CNodeMapFactory ( ) [virtual]

Destroys the node map factory data if all references to the data have been released.

### 10.39.2.3 CNodeMapFactory ( const CNodeMapFactory & )

Creates another reference to the node map factory data.

No data is copied.

### 10.39.2.4 CNodeMapFactory ( EContentType\_t FileType, const GenICam::gcstring & FileName, ECacheUsage\_t CacheUsage = CacheUsage\_Automatic, bool SuppressStringsOnLoad = false )

Creates the node map factory and simply stores the full path to the provided camera description file data.

## Parameters

|    |                              |                                                                                                                                         |
|----|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>FileType</i>              | Defines how the camera description file is stored, e.g. as zipped XML text.                                                             |
| in | <i>FileName</i>              | The full path of the camera description file to process.                                                                                |
| in | <i>CacheUsage</i>            | Defines if and how to use the cache for preprocessed camera description files.                                                          |
| in | <i>SuppressStringsOnLoad</i> | Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint. |

Throws an invalid argument exception if *FileName* is empty. Throws if environment variables in *FileName* cannot be resolved.

## Attention

The given file must be readable until the camera description file data has been released. The [IsCameraDescriptionFileDataReleased\(\)](#) method can be used to check if releasing has been done.

**10.39.2.5 CNodeMapFactory ( EContentType\_t ContentType, const void \* pData, size\_t DataSize, ECacheUsage\_t CacheUsage = CacheUsage\_Automatic, bool SuppressStringsOnLoad = false )**

Creates the node map factory and simply stores the pointer and the size of the provided camera description file data.

## Parameters

|    |                              |                                                                                                                                         |
|----|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>ContentType</i>           | Defines how the camera description file is stored, e.g. as zipped XML text.                                                             |
| in | <i>pData</i>                 | The pointer to the camera description file data.                                                                                        |
| in | <i>DataSize</i>              | The size of the camera description file data.                                                                                           |
| in | <i>CacheUsage</i>            | Defines if and how to use the cache for preprocessed camera description files.                                                          |
| in | <i>SuppressStringsOnLoad</i> | Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint. |

Throws an invalid argument exception if *pData* is NULL or *DataSize* is 0.

## Attention

The given buffer must not be freed or changed until the camera description file data has been released. The [IsCameraDescriptionFileDataReleased\(\)](#) method can be used to check if releasing has been done.

**10.39.2.6 CNodeMapFactory ( const GenICam::gcstring & XmlData, ECacheUsage\_t CacheUsage = CacheUsage\_Automatic, bool SuppressStringsOnLoad = false )**

Creates the node map factory and copies the provided camera description file string.

## Parameters

|    |                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>XmlData</i>               | The camera description file data as XML text. The provided text is copied. You can use the overloaded constructor accepting a buffer to avoid that.<br><br><pre>gcstring cdfData; //... fill cdfData ... CNodeMapFactory factory(ContentType_Xml, cfdData.c_str(), cfdData.size()); // Create the node map. The node map can be destroyed using the IDestroy interface later. INodeMap* pNodeMap = factory.CreateNodeMap(); // The next step is attaching the device port (not shown).</pre> |
| in | <i>CacheUsage</i>            | Defines if and how to use the cache for preprocessed camera description files.                                                                                                                                                                                                                                                                                                                                                                                                               |
| in | <i>SuppressStringsOnLoad</i> | Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint.                                                                                                                                                                                                                                                                                                                                                      |

Throws an invalid argument exception if XmlData is empty.

### 10.39.3 Member Function Documentation

#### 10.39.3.1 void AddInjectionData ( CNodeMapFactory & injectionData )

Adds a node map factory representing a camera description file to inject.

## Parameters

|    |                      |                                                                      |
|----|----------------------|----------------------------------------------------------------------|
| in | <i>injectionData</i> | A node map factory representing a camera description file to inject. |
|----|----------------------|----------------------------------------------------------------------|

The injected files are injected in the order they are added. InjectionData must not be preprocessed. The [IsPreprocessed\(\)](#) method can be used to check if preprocessing has been done before. The cache usage of injection data is ignored.

#### 10.39.3.2 const GenICam::gcstring ApplyStyleSheet ( const GenICam::gcstring & StyleSheetFileName )

Applies a style sheet to the pre-processed node map.

#### 10.39.3.3 static bool ClearCache ( ) [static]

Deletes all preprocessed camera description files from the cache.

#### 10.39.3.4 static INodeMap\* CreateEmptyNodeMap ( ) [static]

Creates an empty node map usable as placeholder, e.g.

if certain features are not supported by a module.

10.39.3.5 **static CNodeDataMap\* CreateNodeDataFromNodeMap ( INodeMap \* pNodeMap )** [static]

10.39.3.6 **INodeMap\* CreateNodeMap ( const GenICam::gcstring & DeviceName = "Device", bool DoReleaseCameraDescriptionFileData = true )**

Creates a node map from the preprocessed memory internal representation of the camera description file(s).

[Preprocess\(\)](#) is automatically called if needed. The preprocess step can be omitted by the factory depending on the cache mode setting when a cache file is available, then the cache file is read and converted directly into a node map. [ReleaseCameraDescriptionFileData\(\)](#) is called if DoReleaseCameraDescriptionFileData is true. This method can be called multiple times to create multiple instances of a node map.

10.39.3.7 **INodeMap\* CreateNodeMap ( CLock & UserProvidedLock, const GenICam::gcstring & DeviceName = "Device", bool DoReleaseCameraDescriptionFileData = true )**

Creates a node map from the preprocessed memory internal representation of the camera description file(s).

[Preprocess\(\)](#) is automatically called if needed. The preprocess step can be omitted by the factory depending on the cache mode setting when a cache file is available, then the cache file is read and converted directly into a node map. [ReleaseCameraDescriptionFileData\(\)](#) is called if DoReleaseCameraDescriptionFileData is true. This method can be called multiple times to create multiple instances of a node map. This method allows to provide an external lock to avoid using too many locks in an application.

#### Attention

The provided lock must not be destroyed before the created node map.

10.39.3.8 **CNodeMapFactory ExtractSubtree ( const GenICam::gcstring & SubTreeRootNodeName, bool doRenameToRoot = false )**

The name of the node that represents the root of the subtree that shall be extracted.

#### Parameters

|    |                            |                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | <i>SubTreeRootNodeName</i> | The root of the branch to extract, e.g. "ChunkData".                                                                                                                                                                                                                                                                                                                                                                     |
| in | <i>doRenameToRoot</i>      | Renames the extracted subtree root node SubTreeRootNodeName to "Root", sets the IsFeature property. <a href="#">Preprocess()</a> is automatically called if needed to create the memory internal representation of the camera description file(s). The preprocessed result can be read from the cache or written to the cache in this step. This depends on the availability of a cache and the used CacheUsage setting. |

10.39.3.9 **void GetNodeStatistics ( NodeStatistics\_t & NodeStatistics )**

10.39.3.10 **void GetSupportedSchemaVersions ( GenICam::gcstring\_vector & SchemaVersions ) const**

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

#### 10.39.3.11 bool IsCameraDescriptionFileDataReleased ( ) const

Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.

#### 10.39.3.12 bool IsEmpty ( ) const

Returns true if nothing is loaded ([IsLoaded\(\)](#)) and no source data is available, e.g.

when the node map factory has been created with the default constructor.

#### 10.39.3.13 bool IsLoaded ( ) const

Can be used to check whether the [LoadAndInject\(\)](#) processing step has been performed.

Returns true if [IsPreprocessed\(\)](#) returns true (Preprocessed Data has been loaded from cache).

#### 10.39.3.14 bool IsPreprocessed ( ) const

Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.

#### 10.39.3.15 void LoadAndInject ( )

Advanced: Loads, Parses, and Injects the camera description files recursively.

The result is a memory internal representation of the camera description file(s), the CNodeDataMap (not part of the public interface).

This step is usually done automatically. Prevents cache read if called manually.

#### 10.39.3.16 CNodeMapFactory& operator= ( const CNodeMapFactory & )

Creates another reference to the assigned node map factory data.

Destroys the "overwritten" node map factory data if all references to the data have been released.

#### 10.39.3.17 void Preprocess ( )

Advanced: Creates the preprocessed memory internal representation of the camera description file(s), the C↔NodeDataMap (not part of the public interface).

This step is usually done automatically. Preprocessed data can be read from the cache or written to the cache in this step. This depends on the availability of a cache and the used CacheUsage setting. By calling this method directly direct cache load is suppressed, see [CreateNodeMap\(\)](#) for more information.



## 10.39.3.18 void ReleaseCameraDescriptionFileData ( )

Advanced: Releases any in constructors provided camera description file data buffers or files.

This step is usually done automatically. All references to added injection data are dropped in this step to free the data. After this step any in constructors provided buffers can be freed or any in constructors given files can be deleted.

## 10.39.3.19 GenICam::gcstring ToString ( ) const

Outputs the pre-processed node map in string form (for debug purpose)

## 10.39.3.20 GenICam::gcstring ToXml ( ) const

Outputs the pre-processed node map in XML form (mainly for debug purpose)

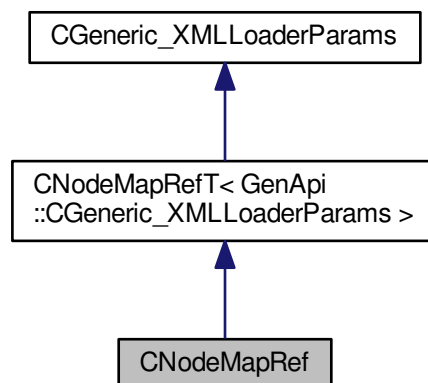
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapFactory.h](#)

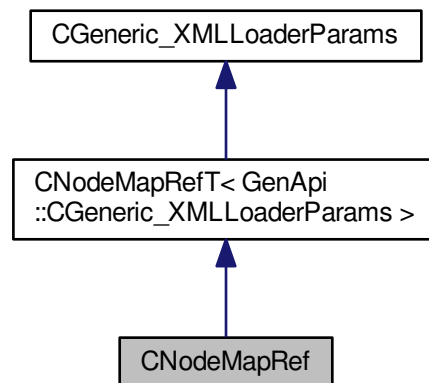
## 10.40 CNodeMapRef Class Reference

Smartpointer for NodeMaps with create function.

Inheritance diagram for CNodeMapRef:



Collaboration diagram for CNodeMapRef:



## Public Member Functions

- `CNodeMapRef` (const `GenICam::gcstring` &DeviceName="Device")  
*Constructor.*
- `CNodeMapRef` (`INodeMap` \*pNodeMap, const `GenICam::gcstring` &DeviceName="Device")  
*Constructor.*
- `CNodeMapRef` (const `CNodeMapRef` &Them)  
*Copy constructor.*
- `CNodeMapRef` & `operator=` (const `CNodeMapRef` &Them)  
*Assignment.*
- `CNodeMapRef` & `operator=` (`INodeMap` \*pNodeMap)  
*Assignment of an INodeMap\*.*

## Additional Inherited Members

### 10.40.1 Detailed Description

Smartpointer for NodeMaps with create function.

#### Note

This class is a simple typedef definition. The class syntax is only used, because Doxygen has to generate a useful documentation.

### 10.40.2 Constructor & Destructor Documentation

#### 10.40.2.1 `CNodeMapRef` ( const `GenICam::gcstring` & *DeviceName* = "Device" ) [inline]

Constructor.

10.40.2.2 **CNodeMapRef** ( **INodeMap** \* *pNodeMap*, const **GenICam::gcstring** & *DeviceName* = "Device" )  
[inline]

Constructor.

10.40.2.3 **CNodeMapRef** ( const **CNodeMapRef** & *Them* ) [inline]

Copy constructor.

### 10.40.3 Member Function Documentation

10.40.3.1 **CNodeMapRef**& operator= ( const **CNodeMapRef** & *Them* ) [inline]

Assignment.

10.40.3.2 **CNodeMapRef**& operator= ( **INodeMap** \* *pNodeMap* ) [inline]

Assignment of an INodeMap\*.

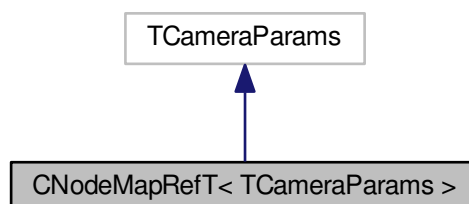
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapRef.h](#)

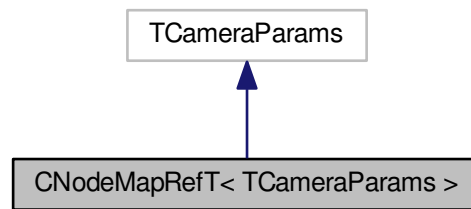
## 10.41 CNodeMapRefT< TCameraParams > Class Template Reference

Smartpointer template for NodeMaps with create function.

Inheritance diagram for CNodeMapRefT< TCameraParams >:



Collaboration diagram for CNodeMapRefT< TCameraParams >:



## Public Member Functions

- `CNodeMapRefT` (const `GenICam::gcstring` &DeviceName="Device")  
*Constructor.*
- `CNodeMapRefT` (`INodeMap` \*pNodeMap, const `GenICam::gcstring` &DeviceName="Device")  
*Constructor.*
- `CNodeMapRefT` (const `CNodeMapRefT` &Them)  
*Copy constructor.*
- `CNodeMapRefT` & `operator=` (const `CNodeMapRefT` &Them)  
*Assignment.*
- `CNodeMapRefT` & `operator=` (`INodeMap` \*pNodeMap)  
*Assignment of an INodeMap\*.*
- virtual `~CNodeMapRefT` ()  
*Destructor.*
- void `_Destroy` ()  
*Destroys the node map.*
- void `_LoadXMLFromFile` (const `GenICam::gcstring` &FileName)  
*Creates the object from a XML file with given file name.*
- void `_LoadXMLFromZIPFile` (const `GenICam::gcstring` &ZipFileName)  
*Creates the object from a ZIP'd XML file with given file name.*
- void `_LoadXMLFromZIPData` (const void \*zipData, size\_t zipSize)  
*Creates the object from a ZIP'd XML file given in a string.*
- void `_LoadXMLFromFileInject` (const `GenICam::gcstring` &TargetFileName, const `GenICam::gcstring` &InjectFileName)  
*Creates the object from a XML target and an inject file with given file name.*
- void `_LoadXMLFromString` (const `GenICam::gcstring` &XMLData)  
*Creates the object from XML data given in a string.*
- void `_LoadXMLFromStringInject` (const `GenICam::gcstring` &TargetXMLDataconst, const `GenICam::gcstring` &InjectXMLData)  
*Creates the object from XML data given in a string with injection.*
- virtual void `_GetSupportedSchemaVersions` (`GenICam::gcstring_vector` &SchemaVersions)  
*Gets a list of supported schema versions.*
- virtual `GenICam::gcstring` `_GetDeviceName` ()  
*Get device name.*
- virtual void `_Poll` (int64\_t ElapsedTime)

*Fires nodes which have a polling time.*

- virtual void [\\_GetNodes](#) ([NodeList\\_t](#) &Nodes)

*Retrieves all nodes in the node map.*

- virtual [INode](#) \* [\\_GetNode](#) (const [GenlCam::gcstring](#) &key)

*Retrieves the node from the central map by name.*

- virtual void [\\_InvalidateNodes](#) ()

*Invalidates all nodes.*

- virtual bool [\\_Connect](#) ([IPort](#) \*pPort, const [GenlCam::gcstring](#) &PortName)

*Connects a port to a port node with given name.*

- virtual bool [\\_Connect](#) ([IPort](#) \*pPort)

*Connects a port to the standard port "Device".*

## Static Public Member Functions

- static bool [\\_ClearXMLCache](#) ()

*Clears the cache of the camera description files.*

## Public Attributes

- [INodeMap](#) \* [\\_Ptr](#)

*Pointer to the [NodeMap](#).*

### 10.41.1 Detailed Description

```
template<class TCameraParams>
class Spinnaker::GenApi::CNodeMapRefT< TCameraParams >
```

Smartpointer template for NodeMaps with create function.

#### Parameters

|                      |                                                                           |
|----------------------|---------------------------------------------------------------------------|
| <i>TCameraParams</i> | The camera specific parameter class (auto generated from camera xml file) |
|----------------------|---------------------------------------------------------------------------|

### 10.41.2 Member Function Documentation

#### 10.41.2.1 static bool [\\_ClearXMLCache](#) ( ) [static]

Clears the cache of the camera description files.

#### 10.41.2.2 virtual bool [\\_Connect](#) ( [IPort](#) \* *pPort*, const [GenlCam::gcstring](#) & *PortName* ) [virtual]

Connects a port to a port node with given name.

10.41.2.3 `virtual bool _Connect ( IPort * pPort ) [virtual]`

Connects a port to the standard port "Device".

10.41.2.4 `virtual GenICam::gcstring _GetDeviceName ( ) [virtual]`

Get device name.

10.41.2.5 `virtual INode* _GetNode ( const GenICam::gcstring & key ) [virtual]`

Retrieves the node from the central map by name.

10.41.2.6 `virtual void _GetNodes ( NodeList_t & Nodes ) [virtual]`

Retrieves all nodes in the node map.

10.41.2.7 `virtual void _GetSupportedSchemaVersions ( GenICam::gcstring_vector & SchemaVersions ) [virtual]`

Gets a list of supported schema versions.

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers. Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only.

10.41.2.8 `virtual void _InvalidateNodes ( ) [virtual]`

Invalidates all nodes.

10.41.2.9 `void _LoadXMLFromFile ( const GenICam::gcstring & FileName )`

Creates the object from a XML file with given file name.

10.41.2.10 `void _LoadXMLFromFileInject ( const GenICam::gcstring & TargetFileName, const GenICam::gcstring & InjectFileName )`

Creates the object from a XML target and an inject file with given file name.

10.41.2.11 `void _LoadXMLFromString ( const GenICam::gcstring & XMLData )`

Creates the object from XML data given in a string.

10.41.2.12 void `_LoadXMLFromStringInject` ( const `GenICam::gcstring` & *TargetXMLData*const, const `GenICam::gcstring` & *InjectXMLData* )

Creates the object from XML data given in a string with injection.

10.41.2.13 void `_LoadXMLFromZIPData` ( const void \* *zipData*, size\_t *zipSize* )

Creates the object from a ZIP'd XML file given in a string.

10.41.2.14 void `_LoadXMLFromZIPFile` ( const `GenICam::gcstring` & *ZipFileName* )

Creates the object from a ZIP'd XML file with given file name.

10.41.2.15 virtual void `_Poll` ( int64\_t *ElapsedTime* ) [virtual]

Fires nodes which have a polling time.

### 10.41.3 Member Data Documentation

10.41.3.1 `INodeMap* _Ptr`

Pointer to the [NodeMap](#).

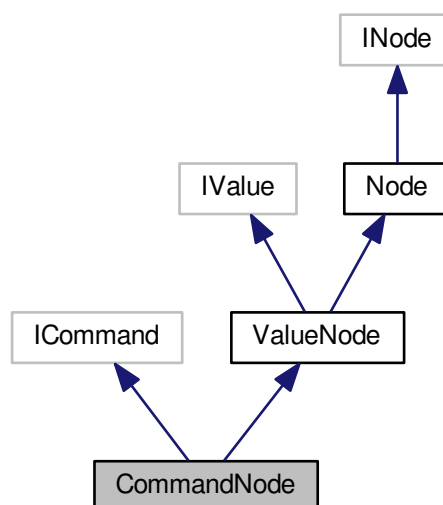
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapRef.h](#)

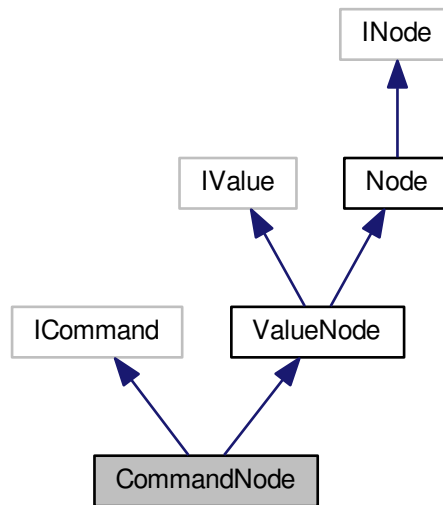
## 10.42 CommandNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for CommandNode:



Collaboration diagram for CommandNode:



## Public Member Functions

- [CommandNode](#) ()
- [CommandNode](#) (std::shared\_ptr< Node::NodeImpl > pCommand)
- virtual [~CommandNode](#) ()
- virtual void [Execute](#) (bool [Verify](#)=true)  
*Execute the command.*
- virtual void [operator\(\)](#) ()  
*Execute the command.*
- virtual bool [IsDone](#) (bool [Verify](#)=true)  
*Query whether the command is executed.*
- virtual void [SetReference](#) ([INode](#) \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.42.1 Detailed Description

[Interface](#) for string properties.



## 10.42.2 Constructor & Destructor Documentation

10.42.2.1 `CommandNode ( )`

10.42.2.2 `CommandNode ( std::shared_ptr< Node::NodeImpl > pCommand )`

10.42.2.3 `virtual ~CommandNode ( ) [virtual]`

## 10.42.3 Member Function Documentation

10.42.3.1 `virtual void Execute ( bool Verify = true ) [virtual]`

Execute the command.

Parameters

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>Verify</i> | Enables AccessMode and Range verification (default = true) |
|---------------|------------------------------------------------------------|

10.42.3.2 `virtual bool IsDone ( bool Verify = true ) [virtual]`

Query whether the command is executed.

Parameters

|               |                                                                                |
|---------------|--------------------------------------------------------------------------------|
| <i>Verify</i> | Enables Range verification (default = false). The AccessMode is always checked |
|---------------|--------------------------------------------------------------------------------|

Returns

True if the Execute command has finished; false otherwise

10.42.3.3 `virtual void operator()( ) [virtual]`

Execute the command.

10.42.3.4 `virtual void SetReference ( INode * pBase ) [virtual]`

overload SetReference for Value

Reimplemented from [ValueNode](#).

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

- include/SpinGenApi/[CommandNode.h](#)

## 10.43 Counter Class Reference

Definition of a simple [Counter](#) class.

### Public Member Functions

- [Counter](#) ()
- unsigned int [GetValue](#) () const
- unsigned int [operator++](#) ()
- unsigned int [operator++](#) (int)
- unsigned int [operator--](#) (int)
- unsigned int [operator--](#) ()
- [operator unsigned int](#) ()
- bool [IsZero](#) ()

### 10.43.1 Detailed Description

Definition of a simple [Counter](#) class.

### 10.43.2 Constructor & Destructor Documentation

10.43.2.1 [Counter](#) ( ) `[inline]`

### 10.43.3 Member Function Documentation

10.43.3.1 unsigned int [GetValue](#) ( ) const `[inline]`

10.43.3.2 bool [IsZero](#) ( ) `[inline]`

10.43.3.3 [operator unsigned int](#) ( ) `[inline]`

10.43.3.4 unsigned int [operator++](#) ( ) `[inline]`

10.43.3.5 unsigned int [operator++](#) ( int ) `[inline]`

10.43.3.6 unsigned int [operator--](#) ( int ) `[inline]`

10.43.3.7 unsigned int [operator--](#) ( ) `[inline]`

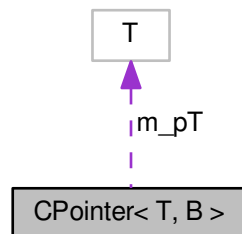
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Counter.h](#)

## 10.44 CPointer< T, B > Class Template Reference

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

Collaboration diagram for CPointer< T, B >:



### Public Member Functions

- [CPointer](#) (void)  
*Default constructor.*
- [CPointer](#) (B \*pB)  
*Constructor from INode pointer type.*
- virtual [~CPointer](#) (void)
- void [operator=](#) (B \*pB)  
*Assign INode Pointer.*
- [operator T \\*](#) (void) const  
*Dereferencing.*
- T & [operator\\*](#) (void) const  
*Dereferencing.*
- T & [operator\(\)](#) (void) const  
*Dereferencing.*
- T \* [operator->](#) (void) const  
*Dereferencing.*
- bool [IsValid](#) () const throw ()  
*true if the pointer is valid*
- [operator bool](#) (void) const throw ()  
*true if the pointer is valid*
- bool [operator==](#) (T \*pT) const  
*pointer equal*
- bool [operator==](#) (const [CPointer](#)< T, B > &rT) const  
*pointer equal*
- bool [operator==](#) (int nMustBeNull) const  
*pointer equal*
- bool [operator!=](#) (const [CPointer](#)< T, B > &rT) const  
*pointer unequal*
- bool [operator!=](#) (T \*pT) const

- pointer unequal*  
• bool `operator!=` (const long int nMustBeNull) const
- pointer unequal*  
• bool `operator!=` (const int nMustBeNull) const
- pointer unequal*

## Protected Attributes

- T \* `m_pT`  
*Underlying raw pointer.*

### 10.44.1 Detailed Description

```
template<class T, class B = IBase>
class Spinnaker::GenApi::CPointer< T, B >
```

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

### 10.44.2 Constructor & Destructor Documentation

10.44.2.1 `CPointer ( void )` `[inline]`

Default constructor.

10.44.2.2 `CPointer ( B * pB )` `[inline]`

Constructor from INode pointer type.

10.44.2.3 `virtual ~CPointer ( void )` `[inline]`, `[virtual]`

### 10.44.3 Member Function Documentation

10.44.3.1 `bool IsValid ( ) const throw )` `[inline]`

true if the pointer is valid

10.44.3.2 `operator bool ( void ) const throw )` `[inline]`

true if the pointer is valid

10.44.3.3 `operator T * ( void ) const` `[inline]`

Dereferencing.

10.44.3.4 `bool operator!=( const CPointer< T, B > & rT ) const` `[inline]`

pointer unequal

10.44.3.5 `bool operator!=( T * pT ) const` `[inline]`

pointer unequal

10.44.3.6 `bool operator!=( const long int nMustBeNull ) const` `[inline]`

pointer unequal

10.44.3.7 `bool operator!=( const int nMustBeNull ) const` `[inline]`

pointer unequal

10.44.3.8 `T& operator()( void ) const` `[inline]`

Dereferencing.

10.44.3.9 `T& operator*( void ) const` `[inline]`

Dereferencing.

10.44.3.10 `T* operator->( void ) const` `[inline]`

Dereferencing.

10.44.3.11 `void operator=( B * pB )` `[inline]`

Assign INode Pointer.

10.44.3.12 `bool operator==( T * pT ) const` `[inline]`

pointer equal

10.44.3.13 `bool operator==( const CPointer< T, B > & rT ) const` `[inline]`

pointer equal



## Public Member Functions

- [CPortImpl](#) ()  
*Constructor.*
- virtual [~CPortImpl](#) ()  
*Destructor.*
- virtual [EAccessMode GetAccessMode](#) () const =0  
*Get the access mode of the node.*
- virtual void [Read](#) (void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))=0  
*Reads a chunk of bytes from the port.*
- virtual void [Write](#) (const void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))=0  
*Writes a chunk of bytes to the port.*
- virtual void [SetPortImpl](#) (IPort \*pPort)  
*Sets pointer the real port implementation; this function may called only once.*
- virtual [EYesNo GetSwapEndianness](#) ()  
*Determines if the port adapter must perform an endianness swap.*
- virtual void [Replay](#) (IPortWriteList \*pPortRecorder, bool [Invalidate](#)=true)  
*sends the commands to the camera.*
- void [InvalidateNode](#) ()

## Protected Attributes

- [CNodePtr m\\_ptrPort](#)  
*Pointer to the node holding a reference to this implementation.*

### 10.45.1 Detailed Description

Standard implementation for a port.

### 10.45.2 Constructor & Destructor Documentation

#### 10.45.2.1 [CPortImpl](#) ( ) [inline]

Constructor.

#### 10.45.2.2 virtual [~CPortImpl](#) ( ) [inline],[virtual]

Destructor.

### 10.45.3 Member Function Documentation

#### 10.45.3.1 virtual [EAccessMode GetAccessMode](#) ( ) const [pure virtual]

Get the access mode of the node.

Driver closed => NI, Driver open => RW, analyzing a struct, RO

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

10.45.3.2 `virtual EYesNo GetSwapEndianness ( ) [inline],[virtual]`

Determines if the port adapter must perform an endianness swap.

10.45.3.3 `void InvalidateNode ( ) [inline]`

10.45.3.4 `virtual void Read ( void * pBuffer, int64_t Address, int64_t Length ) [pure virtual]`

Reads a chunk of bytes from the port.

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

10.45.3.5 `virtual void Replay ( IPortWriteList * pPortRecorder, bool Invalidate = true ) [inline],[virtual]`

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

10.45.3.6 `virtual void SetPortImpl ( IPort * pPort ) [inline],[virtual]`

Sets pointer the real port implementation; this function may called only once.

Reimplemented in [CRegisterPortImpl](#).

10.45.3.7 `virtual void Write ( const void * pBuffer, int64_t Address, int64_t Length ) [pure virtual]`

Writes a chunk of bytes to the port.

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

## 10.45.4 Member Data Documentation

10.45.4.1 `CNodePtr m_ptrPort [protected]`

Pointer to the node holding a reference to this implementation.

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

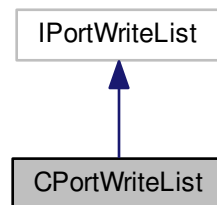
- [include/SpinGenApi/PortImpl.h](#)



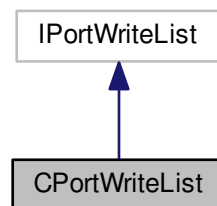
## 10.46 CPortWriteList Class Reference

Container holding a list of port write commands.

Inheritance diagram for CPortWriteList:



Collaboration diagram for CPortWriteList:



### Public Member Functions

- [CPortWriteList](#) ()  
*Constructor.*
- [~CPortWriteList](#) ()  
*Destructor.*
- virtual void [Write](#) (const void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Writes a chunk of bytes to the port.*
- virtual void [Replay](#) (IPort \*pPort)  
*Replays the write command to the given port interface.*
- virtual void [SetCookie](#) (const int64\_t Value)  
*Sets a cookie in case the port implementation want to cache a command list.*
- virtual int64\_t [GetCookie](#) ()  
*Gets the cookie a port implementation may have set for caching a command list.*
- void \* [GetPortWriteListHandle](#) ()

## Protected Attributes

- void \* [m\\_pWriteList](#)

### 10.46.1 Detailed Description

Container holding a list of port write commands.

### 10.46.2 Constructor & Destructor Documentation

#### 10.46.2.1 CPortWriteList ( )

Constructor.

#### 10.46.2.2 ~CPortWriteList ( )

Destructor.

### 10.46.3 Member Function Documentation

#### 10.46.3.1 virtual int64\_t GetCookie ( ) [virtual]

Gets the cookie a port implementation may have set for caching a command list.

#### 10.46.3.2 void\* GetPortWriteListHandle ( )

#### 10.46.3.3 virtual void Replay ( IPort \* *pPort* ) [virtual]

Replays the write command to the given port interface.

#### 10.46.3.4 virtual void SetCookie ( const int64\_t *Value* ) [virtual]

Sets a cookie in case the port implementation want to cache a command list.

#### 10.46.3.5 virtual void Write ( const void \* *pBuffer*, int64\_t *Address*, int64\_t *Length* ) [virtual]

Writes a chunk of bytes to the port.

### 10.46.4 Member Data Documentation

#### 10.46.4.1 void\* m\_pWriteList [protected]

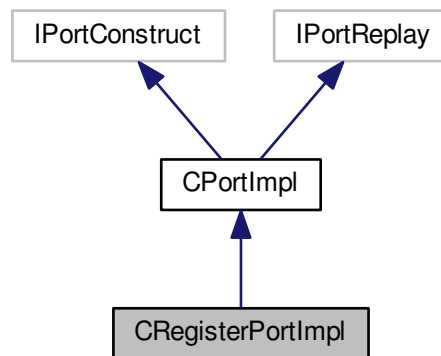
The documentation for this class was generated from the following file:

- include/SpinGenApi/PortWriteList.h

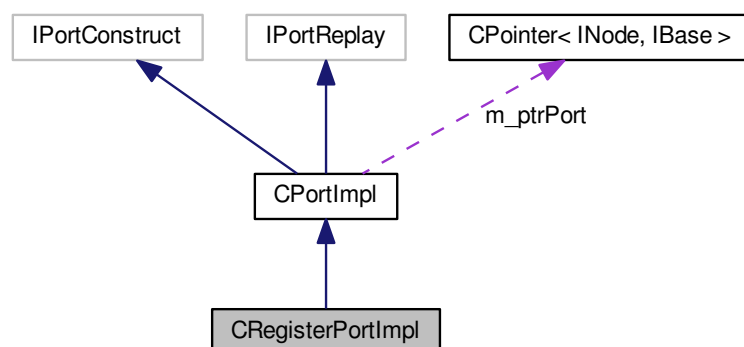
## 10.47 CRegisterPortImpl Class Reference

Standard implementation for a port using a register based transport layer.

Inheritance diagram for CRegisterPortImpl:



Collaboration diagram for CRegisterPortImpl:



## Public Member Functions

- [CRegisterPortImpl](#) (int MaxNumQuadlets=1, bool TransportLayerSwapsEndianness=false)  
*Constructor.*
- virtual [~CRegisterPortImpl](#) ()  
*Destructor.*
- virtual [EAccessMode GetAccessMode](#) () const =0  
*Get the access mode of the node.*
- virtual void [ReadRegister](#) (uint32\_t \*pRegisters, int64\_t [Address](#), int64\_t [Length](#))=0  
*Reads an array of quadlets from the port.*
- virtual void [WriteRegister](#) (const uint32\_t \*pRegisters, int64\_t [Address](#), int64\_t [Length](#))=0  
*Writes an array of quadlets to the port.*
- virtual void [Read](#) (void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Reads a chunk of bytes from the port.*
- virtual void [Write](#) (const void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Writes a chunk of bytes to the port.*
- virtual void [SetPortImpl](#) (IPort \*pPort)  
*Sets pointer the real port implementation; this function may called only once.*

## Additional Inherited Members

### 10.47.1 Detailed Description

Standard implementation for a port using a register based transport layer.

### 10.47.2 Constructor & Destructor Documentation

10.47.2.1 [CRegisterPortImpl](#) ( int *MaxNumQuadlets* = 1, bool *TransportLayerSwapsEndianness* = false ) [inline]

Constructor.

10.47.2.2 virtual [~CRegisterPortImpl](#) ( ) [inline], [virtual]

Destructor.

### 10.47.3 Member Function Documentation

10.47.3.1 virtual [EAccessMode GetAccessMode](#) ( ) const [pure virtual]

Get the access mode of the node.

Driver closed => NI, Driver open => RW, analyzing a struct, RO

Implements [CPortImpl](#).

10.47.3.2 `virtual void Read ( void * pBuffer, int64_t Address, int64_t Length )` `[inline],[virtual]`

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

10.47.3.3 `virtual void ReadRegister ( uint32_t * pRegisters, int64_t Address, int64_t Length )` `[pure virtual]`

Reads an array of quadlets from the port.

10.47.3.4 `virtual void SetPortImpl ( IPort * pPort )` `[inline],[virtual]`

Sets pointer the real port implementation; this function may called only once.

Reimplemented from [CPortImpl](#).

10.47.3.5 `virtual void Write ( const void * pBuffer, int64_t Address, int64_t Length )` `[inline],[virtual]`

Writes a chunk of bytes to the port.

Implements [CPortImpl](#).

10.47.3.6 `virtual void WriteRegister ( const uint32_t * pRegisters, int64_t Address, int64_t Length )` `[pure virtual]`

Writes an array of quadlets to the port.

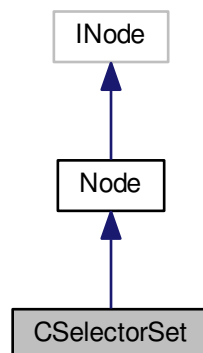
The documentation for this class was generated from the following file:

- `include/SpinGenApi/`[RegisterPortImpl.h](#)

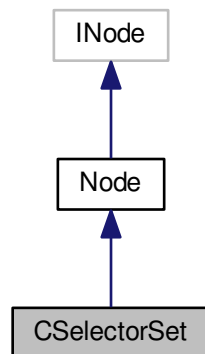
## 10.48 CSelectorSet Class Reference

The set of selectors selecting a given node.

Inheritance diagram for CSelectorSet:



Collaboration diagram for CSelectorSet:



## Public Member Functions

- [CSelectorSet](#) ([IBase](#) \*pBase)  
*Constructor.*
- [~CSelectorSet](#) ()  
*Destructor.*
- bool [IsEmpty](#) ()  
*returns true if no selectors are present*
- virtual bool [SetFirst](#) ()
- virtual bool [SetNext](#) (bool Tick=true)
- virtual void [Restore](#) ()
- virtual [GenICam::gcstring ToString](#) ()
- virtual void [GetSelectorList](#) (FeatureList\_t &SelectorList, bool Incremental=false)

## Additional Inherited Members

### 10.48.1 Detailed Description

The set of selectors selecting a given node.

### 10.48.2 Constructor & Destructor Documentation

#### 10.48.2.1 CSelectorSet ( IBase \* pBase )

Constructor.

## Parameters

|              |                                      |
|--------------|--------------------------------------|
| <i>pBase</i> | Feature selected by the selector set |
|--------------|--------------------------------------|

## 10.48.2.2 ~CSelectorSet ( )

Destructor.

## 10.48.3 Member Function Documentation

10.48.3.1 virtual void GetSelectorList ( FeatureList\_t & *SelectorList*, bool *Incremental* = false ) [virtual]

10.48.3.2 bool IsEmpty ( )

returns true if no selectors are present

10.48.3.3 virtual void Restore ( ) [virtual]

10.48.3.4 virtual bool SetFirst ( ) [virtual]

10.48.3.5 virtual bool SetNext ( bool *Tick* = true ) [virtual]

10.48.3.6 virtual GenICam::gcstring ToString ( ) [virtual]

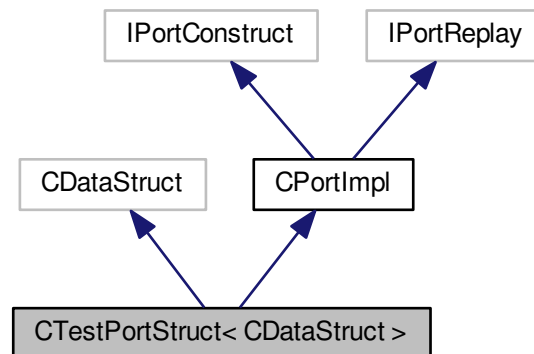
The documentation for this class was generated from the following file:

- include/SpinGenApi/[SelectorSet.h](#)

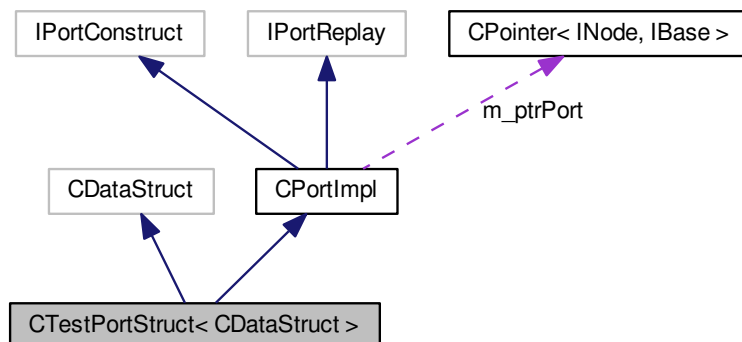
## 10.49 CTestPortStruct&lt; CDataStruct &gt; Class Template Reference

Implements a register spaces based on a C++ struct.

Inheritance diagram for CTestPortStruct< CDataStruct >:



Collaboration diagram for CTestPortStruct< CDataStruct >:



## Public Member Functions

- [CTestPortStruct](#) (int64\_t BaseAddress=0)
- virtual [EAccessMode](#) [GetAccessMode](#) () const  
*Get the access mode of the node.*
- virtual [EInterfaceType](#) [GetPrincipallInterfaceType](#) () const  
*Get the type of the main interface of a node.*
- virtual void [Read](#) (void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Reads a chunk of bytes from the port.*
- virtual void [Write](#) (const void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Writes a chunk of bytes to the port.*
- void [MemSet](#) (const char FillValue)



- void [ResetStatistics](#) ()  
*Resets the read/write statistics.*
- int64\_t [GetNumReads](#) ()  
*Returns the number of reads since lastReset Statistics.*
- int64\_t [GetNumWrites](#) ()  
*Returns the number of writes since lastReset Statistics.*

## Protected Attributes

- int64\_t [m\\_NumReads](#)  
*Number of reads since last reset.*
- int64\_t [m\\_NumWrites](#)  
*Number of writes since last reset.*
- int64\_t [m\\_BaseAddress](#)  
*the base address used for the struct*

### 10.49.1 Detailed Description

```
template<class CDataStruct>
class Spinnaker::GenApi::CTestPortStruct< CDataStruct >
```

Implements a register spaces based on a C++ struct.

### 10.49.2 Constructor & Destructor Documentation

10.49.2.1 [CTestPortStruct](#) ( int64\_t *BaseAddress* = 0 ) `[inline]`

### 10.49.3 Member Function Documentation

10.49.3.1 [virtual EAccessMode GetAccessMode](#) ( ) `const` `[inline]`, `[virtual]`

Get the access mode of the node.

Implements [CPortImpl](#).

10.49.3.2 [int64\\_t GetNumReads](#) ( ) `[inline]`

Returns the number of reads since lastReset Statistics.

10.49.3.3 [int64\\_t GetNumWrites](#) ( ) `[inline]`

Returns the number of writes since lastReset Statistics.

10.49.3.4 `virtual EInterfaceType GetPrincipalInterfaceType ( ) const` `[inline],[virtual]`

Get the type of the main interface of a node.

10.49.3.5 `void MemSet ( const char FillValue )` `[inline]`

10.49.3.6 `virtual void Read ( void * pBuffer, int64_t Address, int64_t Length )` `[inline],[virtual]`

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

10.49.3.7 `void ResetStatistics ( )` `[inline]`

Resets the read/write statistics.

10.49.3.8 `virtual void Write ( const void * pBuffer, int64_t Address, int64_t Length )` `[inline],[virtual]`

Writes a chunk of bytes to the port.

Implements [CPortImpl](#).

## 10.49.4 Member Data Documentation

10.49.4.1 `int64_t m_BaseAddress` `[protected]`

the base address used for the struct

10.49.4.2 `int64_t m_NumReads` `[protected]`

Number of reads since last reset.

10.49.4.3 `int64_t m_NumWrites` `[protected]`

Number of writes since last reset.

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

- `include/SpinGenApi/StructPort.h`

## 10.50 DCAM\_CHECKSUM Struct Reference

### Public Attributes

- `uint32_t` [CRCChecksum](#)

### 10.50.1 Member Data Documentation

#### 10.50.1.1 uint32\_t CRCChecksum

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

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

## 10.51 DCAM\_CHUNK\_TRAILER Struct Reference

### Public Attributes

- SPIN\_GUID [ChunkID](#)
- uint32\_t [ChunkLength](#)
- uint32\_t [InverseChunkLength](#)

### 10.51.1 Member Data Documentation

#### 10.51.1.1 SPIN\_GUID ChunkID

#### 10.51.1.2 uint32\_t ChunkLength

#### 10.51.1.3 uint32\_t InverseChunkLength

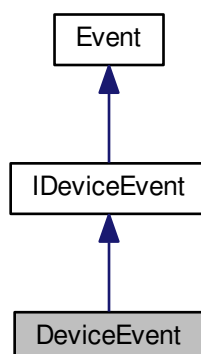
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

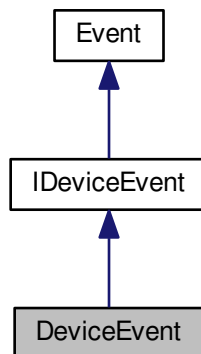
## 10.52 DeviceEvent Class Reference

A handler to device events.

Inheritance diagram for DeviceEvent:



Collaboration diagram for DeviceEvent:



## Public Member Functions

- [DeviceEvent](#) ()  
*Default constructor.*
- virtual [~DeviceEvent](#) ()  
*Virtual destructor.*
- virtual void [OnDeviceEvent](#) ([Spinnaker::GenICam::gcstring](#) eventName)=0  
*Device event callback.*
- uint64\_t [GetDeviceEventId](#) () const  
*Get the ID of the device event.*
- [GenICam::gcstring](#) [GetDeviceEventName](#) () const  
*Get the name of the device event.*

## Protected Member Functions

- [DeviceEvent](#) & [operator=](#) (const [DeviceEvent](#) &)  
*Assignment operator.*

## Additional Inherited Members

### 10.52.1 Detailed Description

A handler to device events.

### 10.52.2 Constructor & Destructor Documentation

#### 10.52.2.1 [DeviceEvent](#) ( )

Default constructor.

10.52.2.2 `virtual ~DeviceEvent ( ) [virtual]`

Virtual destructor.

### 10.52.3 Member Function Documentation

10.52.3.1 `uint64_t GetDeviceEventId ( ) const [virtual]`

Get the ID of the device event.

#### Returns

The device event ID

Implements [IDeviceEvent](#).

10.52.3.2 `GenICam::gcstring GetDeviceEventName ( ) const [virtual]`

Get the name of the device event.

#### Returns

The device event name

Implements [IDeviceEvent](#).

10.52.3.3 `virtual void OnDeviceEvent ( Spinnaker::GenICam::gcstring eventName ) [pure virtual]`

Device event callback.

#### Parameters

|                  |                       |
|------------------|-----------------------|
| <i>eventName</i> | The name of the event |
|------------------|-----------------------|

Implements [IDeviceEvent](#).

10.52.3.4 `DeviceEvent& operator= ( const DeviceEvent & ) [protected]`

Assignment operator.

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

- `include/DeviceEvent.h`

## 10.53 `double_autovector_t` Class Reference

Vector of doubles with reference counting.

### Public Member Functions

- `double_autovector_t` ()
- `double_autovector_t` (const `double_autovector_t` &obj)
- `double_autovector_t` (size\_t n)
- virtual `~double_autovector_t` (void)
- `double_autovector_t` & `operator=` (const `double_autovector_t` &obj)
- void `operator delete` (void \*pWhere)
- void \* `operator new` (size\_t uiSize)
- double & `operator[]` (size\_t uiIndex)
- const double & `operator[]` (size\_t uiIndex) const
- size\_t `size` () const

### Protected Attributes

- std::vector< double > \* `_pv`
- ATOMIC\_VARIABLE \* `_pCount`

#### 10.53.1 Detailed Description

Vector of doubles with reference counting.

#### 10.53.2 Constructor & Destructor Documentation

10.53.2.1 `double_autovector_t` ( )

10.53.2.2 `double_autovector_t` ( const `double_autovector_t` & *obj* )

10.53.2.3 `double_autovector_t` ( size\_t *n* ) [explicit]

10.53.2.4 virtual `~double_autovector_t` ( void ) [virtual]

#### 10.53.3 Member Function Documentation

10.53.3.1 void `operator delete` ( void \* *pWhere* )

10.53.3.2 void\* `operator new` ( size\_t *uiSize* )

10.53.3.3 `double_autovector_t`& `operator=` ( const `double_autovector_t` & *obj* )

10.53.3.4 double& `operator[]` ( size\_t *uiIndex* )

10.53.3.5 `const double& operator[] ( size_t uiIndex ) const`

10.53.3.6 `size_t size ( ) const`

## 10.53.4 Member Data Documentation

10.53.4.1 `ATOMIC_VARIABLE* _pCount` `[protected]`

10.53.4.2 `std::vector<double>* _pv` `[protected]`

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

- `include/SpinGenApi/Autovector.h`

## 10.54 EAccessModeClass Class Reference

Holds conversion methods for the access mode enumeration.

### Static Public Member Functions

- static bool `FromString` (const `GenlCam::gcstring` &ValueStr, `EAccessMode` \*pValue)  
*Converts a string to enum value.*
- static void `ToString` (`GenlCam::gcstring` &ValueStr, `EAccessMode` \*pValue)  
*Converts a string to an int32\_t property.*
- static `GenlCam::gcstring ToString` (`EAccessMode` Value)  
*Converts a string to an int32\_t property.*

### 10.54.1 Detailed Description

Holds conversion methods for the access mode enumeration.

### 10.54.2 Member Function Documentation

10.54.2.1 `static bool FromString ( const GenlCam::gcstring & ValueStr, EAccessMode * pValue )` `[static]`

Converts a string to enum value.

10.54.2.2 `static void ToString ( GenlCam::gcstring & ValueStr, EAccessMode * pValue )` `[static]`

Converts a string to an int32\_t property.

10.54.2.3 `static GenICam::gcstring ToString ( EAccessMode Value ) [static]`

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.55 ECachingModeClass Class Reference

Holds conversion methods for the caching mode enumeration.

### Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ECachingMode](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ECachingMode](#) \*pValue)
- static [GenICam::gcstring ToString](#) ([ECachingMode](#) Value)  
*Converts a string to an int32\_t property.*

### 10.55.1 Detailed Description

Holds conversion methods for the caching mode enumeration.

### 10.55.2 Member Function Documentation

10.55.2.1 `static bool FromString ( const GenICam::gcstring & ValueStr, ECachingMode * pValue ) [static]`

Converts a string to enum value.

10.55.2.2 `static void ToString ( GenICam::gcstring & ValueStr, ECachingMode * pValue ) [static]`

10.55.2.3 `static GenICam::gcstring ToString ( ECachingMode Value ) [static]`

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.56 EDisplayNotationClass Class Reference

Holds conversion methods for the notation type of floats.



## Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EDisplayNotation](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EDisplayNotation](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) ([EDisplayNotation](#) Value)  
*Converts a string to an int32\_t property.*

### 10.56.1 Detailed Description

Holds conversion methods for the notation type of floats.

### 10.56.2 Member Function Documentation

10.56.2.1 static bool [FromString](#) ( const [GenICam::gcstring](#) & *ValueStr*, [EDisplayNotation](#) \* *pValue* ) [static]

Converts a string to enum value.

10.56.2.2 static void [ToString](#) ( [GenICam::gcstring](#) & *ValueStr*, [EDisplayNotation](#) \* *pValue* ) [static]

Converts a string to an int32\_t property.

10.56.2.3 static [GenICam::gcstring](#) [ToString](#) ( [EDisplayNotation](#) *Value* ) [static]

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.57 EEndianessClass Class Reference

Holds conversion methods for the endianness enumeration.

## Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EEndianess](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EEndianess](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) ([EEndianess](#) Value)  
*Converts a string to an int32\_t property.*

### 10.57.1 Detailed Description

Holds conversion methods for the endianess enumeration.

### 10.57.2 Member Function Documentation

10.57.2.1 `static bool FromString ( const GenlCam::gcstring & ValueStr, EEndianess * pValue ) [static]`

Converts a string to enum value.

10.57.2.2 `static void ToString ( GenlCam::gcstring & ValueStr, EEndianess * pValue ) [static]`

Converts a string to an int32\_t property.

10.57.2.3 `static GenlCam::gcstring ToString ( EEndianess Value ) [static]`

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.58 EGenApiSchemaVersionClass Class Reference

helper class converting EGenApiSchemaVersion from and to string

### Static Public Member Functions

- static bool [FromString](#) (const [GenlCam::gcstring](#) &ValueStr, [EGenApiSchemaVersion](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenlCam::gcstring](#) &ValueStr, [EGenApiSchemaVersion](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenlCam::gcstring](#) [ToString](#) ([EGenApiSchemaVersion](#) Value)  
*Converts a string to an int32\_t property.*

### 10.58.1 Detailed Description

helper class converting EGenApiSchemaVersion from and to string

## 10.58.2 Member Function Documentation

10.58.2.1 static bool FromString ( const GenICam::gcstring & ValueStr, EGenApiSchemaVersion \* pValue )  
[static]

Converts a string to enum value.

10.58.2.2 static void ToString ( GenICam::gcstring & ValueStr, EGenApiSchemaVersion \* pValue ) [static]

Converts a string to an int32\_t property.

10.58.2.3 static GenICam::gcstring ToString ( EGenApiSchemaVersion Value ) [static]

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.59 EInputDirectionClass Class Reference

Holds conversion methods for the notation type of floats.

### Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EInputDirection](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EInputDirection](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) ([EInputDirection](#) Value)  
*Converts a string to an int32\_t property.*

## 10.59.1 Detailed Description

Holds conversion methods for the notation type of floats.

## 10.59.2 Member Function Documentation

10.59.2.1 static bool FromString ( const GenICam::gcstring & ValueStr, EInputDirection \* pValue ) [static]

Converts a string to enum value.

10.59.2.2 `static void ToString ( GenICam::gcstring & ValueStr, EInputDirection * pValue ) [static]`

Converts a string to an int32\_t property.

10.59.2.3 `static GenICam::gcstring ToString ( EInputDirection Value ) [static]`

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.60 ENameSpaceClass Class Reference

Holds conversion methods for the namespace enumeration.

### Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ENameSpace](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ENameSpace](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) ([ENameSpace](#) Value)  
*Converts a string to an int32\_t property.*

### 10.60.1 Detailed Description

Holds conversion methods for the namespace enumeration.

### 10.60.2 Member Function Documentation

10.60.2.1 `static bool FromString ( const GenICam::gcstring & ValueStr, ENameSpace * pValue ) [static]`

Converts a string to enum value.

10.60.2.2 `static void ToString ( GenICam::gcstring & ValueStr, ENameSpace * pValue ) [static]`

Converts a string to an int32\_t property.

10.60.2.3 `static GenICam::gcstring ToString ( ENamespace Value )` `[static]`

Converts a string to an `int32_t` property.

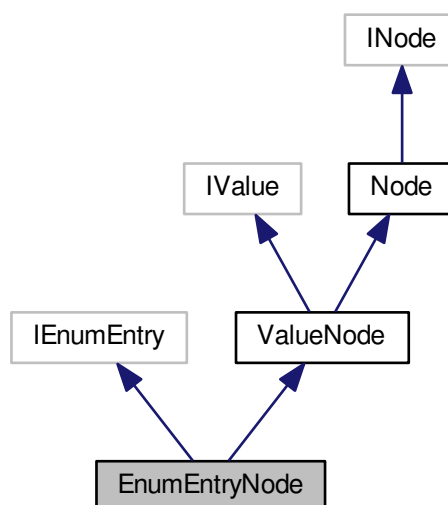
The documentation for this class was generated from the following file:

- `include/SpinGenApi/EnumClasses.h`

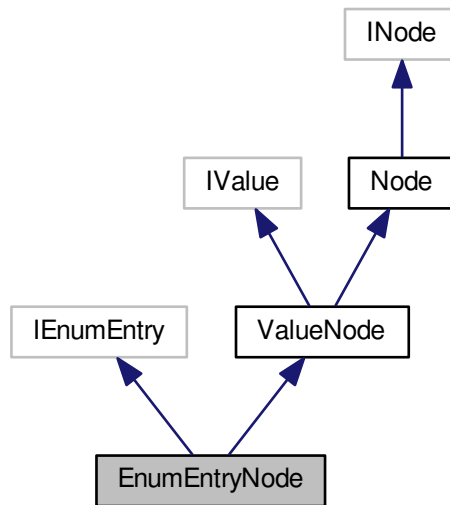
## 10.61 EnumEntryNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for EnumEntryNode:



Collaboration diagram for EnumEntryNode:



## Public Member Functions

- [EnumEntryNode](#) ()
- [EnumEntryNode](#) (std::shared\_ptr< Node::NodeImpl > pEnumEntry)
- virtual [~EnumEntryNode](#) ()
- virtual int64\_t [GetValue](#) ()  
*Get numeric enum value.*
- virtual [GenICam::gcstring](#) [GetSymbolic](#) () const  
*Get symbolic enum value.*
- virtual double [GetNumericValue](#) ()  
*Get double number associated with the entry.*
- virtual bool [IsSelfClearing](#) ()  
*Indicates if the corresponding EnumEntry is self clearing.*
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for EnumEntry*

## Additional Inherited Members

### 10.61.1 Detailed Description

[Interface](#) for string properties.

## 10.61.2 Constructor & Destructor Documentation

10.61.2.1 `EnumEntryNode ( )`

10.61.2.2 `EnumEntryNode ( std::shared_ptr< Node::NodeImpl > pEnumEntry )`

10.61.2.3 `virtual ~EnumEntryNode ( )` [virtual]

## 10.61.3 Member Function Documentation

10.61.3.1 `virtual double GetNumericValue ( )` [virtual]

Get double number associated with the entry.

10.61.3.2 `virtual GenlCam::gcstring GetSymbolic ( ) const` [virtual]

Get symbolic enum value.

10.61.3.3 `virtual int64_t GetValue ( )` [virtual]

Get numeric enum value.

10.61.3.4 `virtual bool IsSelfClearing ( )` [virtual]

Indicates if the corresponding EnumEntry is self clearing.

10.61.3.5 `virtual void SetReference ( INode * pBase )` [virtual]

overload SetReference for EnumEntry

Reimplemented from [ValueNode](#).

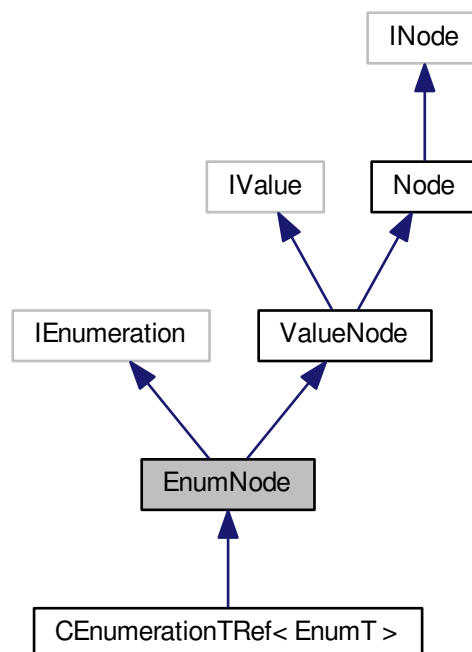
The documentation for this class was generated from the following file:

- `include/SpinGenApi/EnumEntryNode.h`

## 10.62 EnumNode Class Reference

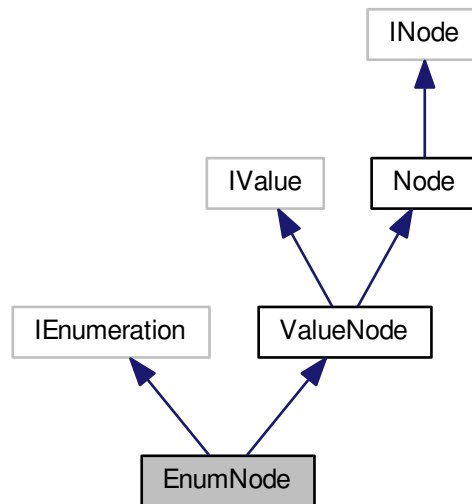
[Interface](#) for string properties.

Inheritance diagram for EnumNode:





Collaboration diagram for EnumNode:



## Public Member Functions

- [EnumNode](#) ()
- [EnumNode](#) (std::shared\_ptr< Node::NodeImpl > pEnumeration)
- virtual [~EnumNode](#) ()
- virtual void [GetSymbolics](#) (StringList\_t &Symbolics)  
*Get list of symbolic Values.*
- virtual void [GetEntries](#) (NodeList\_t &Entries)  
*Get list of entry nodes.*
- virtual [IEnumeration](#) & [operator=](#) (const [GenICam::gcstring](#) &ValueStr)  
*Set string node value.*
- virtual void [SetIntValue](#) (int64\_t Value, bool [Verify](#)=true)  
*Set integer node value.*
- virtual [GenICam::gcstring](#) [operator\\*](#) ()  
*Get string node value.*
- virtual int64\_t [GetIntValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)  
*Get integer node value.*
- virtual [IEnumEntry](#) \* [GetEntryByName](#) (const [GenICam::gcstring](#) &Symbolic)  
*Get an entry node by name.*
- virtual [IEnumEntry](#) \* [GetEntry](#) (const int64\_t IntValue)  
*Get an entry node by its IntValue.*
- virtual [IEnumEntry](#) \* [GetCurrentEntry](#) (bool [Verify](#)=false, bool IgnoreCache=false)  
*Get the current entry.*
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Enumeration*

## Protected Attributes

- `std::shared_ptr< Node::NodeImpl > m_pEnumeration`

### 10.62.1 Detailed Description

[Interface](#) for string properties.

### 10.62.2 Constructor & Destructor Documentation

10.62.2.1 `EnumNode ( )`

10.62.2.2 `EnumNode ( std::shared_ptr< Node::NodeImpl > pEnumeration )`

10.62.2.3 `virtual ~EnumNode ( ) [virtual]`

### 10.62.3 Member Function Documentation

10.62.3.1 `virtual IEnumEntry* GetCurrentEntry ( bool Verify = false, bool IgnoreCache = false ) [virtual]`

Get the current entry.

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.62.3.2 `virtual void GetEntries ( NodeList_t & Entries ) [virtual]`

Get list of entry nodes.

10.62.3.3 `virtual IEnumEntry* GetEntry ( const int64_t IntValue ) [virtual]`

Get an entry node by its IntValue.

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.62.3.4 `virtual IEnumEntry* GetEntryByName ( const GenICam::gcstring & Symbolic ) [virtual]`

Get an entry node by name.

10.62.3.5 `virtual int64_t GetIntValue ( bool Verify = false, bool IgnoreCache = false ) [virtual]`

Get integer node value.

## Parameters

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| <i>Verify</i>      | Enables Range verification (default = false). The AccessMode is always checked |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false)                |

## Returns

The value read

10.62.3.6 virtual void GetSymbolics ( [StringList\\_t](#) & *Symbolics* ) [virtual]

Get list of symbolic Values.

10.62.3.7 virtual [GenICam::gcstring](#) operator\*( ) [virtual]

Get string node value.

10.62.3.8 virtual [IEnumeration&](#) operator= ( const [GenICam::gcstring](#) & *ValueStr* ) [virtual]

Set string node value.

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.62.3.9 virtual void SetIntValue ( [int64\\_t](#) *Value*, bool *Verify* = true ) [virtual]

Set integer node value.

## Parameters

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>Value</i>  | The value to set                                           |
| <i>Verify</i> | Enables AccessMode and Range verification (default = true) |

10.62.3.10 virtual void SetReference ( [INode](#) \* *pBase* ) [virtual]

overload SetReference for Enumeration

Reimplemented from [ValueNode](#).

Reimplemented in [CEnumerationTRef< EnumT >](#).

## 10.62.4 Member Data Documentation

10.62.4.1 [std::shared\\_ptr<Node::NodeImpl>](#) m\_pEnumeration [protected]

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

- include/SpinGenApi/[EnumNode.h](#)

## 10.63 ERepresentationClass Class Reference

Holds conversion methods for the representation enumeration.

### Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ERepresentation](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ERepresentation](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) ([ERepresentation](#) Value)  
*Converts a string to an int32\_t property.*

### 10.63.1 Detailed Description

Holds conversion methods for the representation enumeration.

### 10.63.2 Member Function Documentation

10.63.2.1 static bool [FromString](#) ( const [GenICam::gcstring](#) & *ValueStr*, [ERepresentation](#) \* *pValue* ) [static]

Converts a string to enum value.

10.63.2.2 static void [ToString](#) ( [GenICam::gcstring](#) & *ValueStr*, [ERepresentation](#) \* *pValue* ) [static]

Converts a string to an int32\_t property.

10.63.2.3 static [GenICam::gcstring](#) [ToString](#) ( [ERepresentation](#) *Value* ) [static]

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.64 ESignClass Class Reference

Holds conversion methods for the sign enumeration.

### Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ESign](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ESign](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) ([ESign](#) Value)  
*Converts a string to an int32\_t property.*

#### 10.64.1 Detailed Description

Holds conversion methods for the sign enumeration.

#### 10.64.2 Member Function Documentation

10.64.2.1 static bool [FromString](#) ( const [GenICam::gcstring](#) & *ValueStr*, [ESign](#) \* *pValue* ) [static]

Converts a string to enum value.

10.64.2.2 static void [ToString](#) ( [GenICam::gcstring](#) & *ValueStr*, [ESign](#) \* *pValue* ) [static]

Converts a string to an int32\_t property.

10.64.2.3 static [GenICam::gcstring](#) [ToString](#) ( [ESign](#) *Value* ) [static]

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.65 ESlopeClass Class Reference

Holds conversion methods for the converter formulas.

### Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [ESlope](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [ESlope](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) ([ESlope](#) Value)  
*Converts a string to an int32\_t property.*

### 10.65.1 Detailed Description

Holds conversion methods for the converter formulas.

### 10.65.2 Member Function Documentation

10.65.2.1 `static bool FromString ( const GenICam::gcstring & ValueStr, ESlope * pValue )` `[static]`

Converts a string to enum value.

10.65.2.2 `static void ToString ( GenICam::gcstring & ValueStr, ESlope * pValue )` `[static]`

Converts a string to an int32\_t property.

10.65.2.3 `static GenICam::gcstring ToString ( ESlope Value )` `[static]`

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.66 EStandardNameSpaceClass Class Reference

Holds conversion methods for the standard namespace enumeration.

### Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EStandardNameSpace](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EStandardNameSpace](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) ([EStandardNameSpace](#) Value)  
*Converts a string to an int32\_t property.*

### 10.66.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

## 10.66.2 Member Function Documentation

10.66.2.1 `static bool FromString ( const GenICam::gcstring & ValueStr, EStandardNameSpace * pValue )`  
`[static]`

Converts a string to enum value.

10.66.2.2 `static void ToString ( GenICam::gcstring & ValueStr, EStandardNameSpace * pValue )` `[static]`

Converts a string to an int32\_t property.

10.66.2.3 `static GenICam::gcstring ToString ( EStandardNameSpace Value )` `[static]`

Converts a string to an int32\_t property.

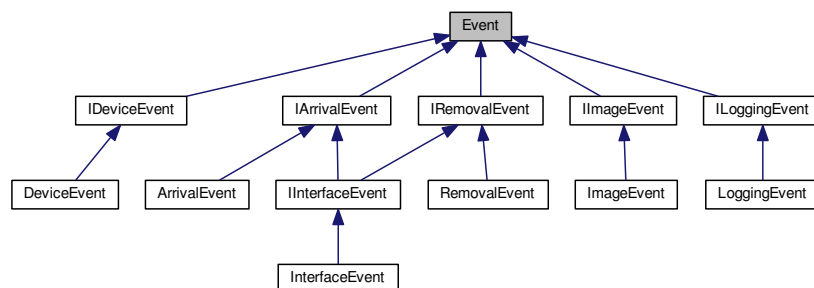
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

## 10.67 Event Class Reference

The base class for all event types.

Inheritance diagram for Event:



### Public Member Functions

- `virtual ~Event ()`  
*Virtual Destructor.*
- `void SetEventType (EventType eventType)`  
*Sets the event type.*
- `EventType GetEventType ()`  
*Gets the event type.*
- `const uint8_t * GetEventPayloadData ()`  
*Gets the event payload data.*
- `const size_t GetEventPayloadDataSize ()`  
*Gets the event payload data size.*

## Protected Member Functions

- [Event](#) ()
- [Event](#) & [operator=](#) (const [Event](#) &)
- void [SetEventPayload](#) (uint8\_t \*offset, size\_t length)

## Protected Attributes

- EventData \* [m\\_pEventData](#)

## Friends

- class [EventProcessor](#)
- class [IDataStream](#)
- class [Stream](#)

### 10.67.1 Detailed Description

The base class for all event types.

### 10.67.2 Constructor & Destructor Documentation

#### 10.67.2.1 virtual ~Event ( ) [virtual]

Virtual Destructor.

#### 10.67.2.2 Event ( ) [protected]

### 10.67.3 Member Function Documentation

#### 10.67.3.1 const uint8\_t\* GetEventPayloadData ( )

Gets the event payload data.

##### Returns

The event payload data

#### 10.67.3.2 const size\_t GetEventPayloadDataSize ( )

Gets the event payload data size.

##### Returns

The event payload data size



**10.67.3.3 EventType GetEventType ( )**

Gets the event type.

**Returns**

The event type

**10.67.3.4 Event& operator= ( const Event & ) [protected]****10.67.3.5 void SetEventPayload ( uint8\_t\* *offset*, size\_t *length* ) [protected]****10.67.3.6 void SetEventType ( EventType *eventType* )**

Sets the event type.

**Parameters**

|                  |                |
|------------------|----------------|
| <i>eventType</i> | The event type |
|------------------|----------------|

**10.67.4 Friends And Related Function Documentation****10.67.4.1 friend class EventProcessor [friend]****10.67.4.2 friend class IDataStream [friend]****10.67.4.3 friend class Stream [friend]****10.67.5 Member Data Documentation****10.67.5.1 EventData\* m\_pEventData [protected]**

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

- [include/Event.h](#)

**10.68 EVisibilityClass Class Reference**

Holds conversion methods for the visibility enumeration.

## Static Public Member Functions

- static bool [FromString](#) (const [GenlCam::gcstring](#) &ValueStr, [EVisibility](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenlCam::gcstring](#) &ValueStr, [EVisibility](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenlCam::gcstring](#) [ToString](#) ([EVisibility](#) Value)  
*Converts a string to an int32\_t property.*

### 10.68.1 Detailed Description

Holds conversion methods for the visibility enumeration.

### 10.68.2 Member Function Documentation

10.68.2.1 static bool [FromString](#) ( const [GenlCam::gcstring](#) & *ValueStr*, [EVisibility](#) \* *pValue* ) [static]

Converts a string to enum value.

10.68.2.2 static void [ToString](#) ( [GenlCam::gcstring](#) & *ValueStr*, [EVisibility](#) \* *pValue* ) [static]

Converts a string to an int32\_t property.

10.68.2.3 static [GenlCam::gcstring](#) [ToString](#) ( [EVisibility](#) *Value* ) [static]

Converts a string to an int32\_t property.

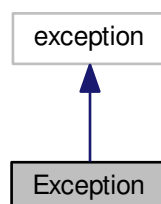
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

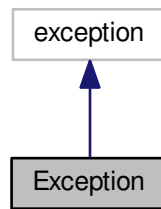
## 10.69 Exception Class Reference

The [Exception](#) object represents an error that is returned from the library.

Inheritance diagram for Exception:



Collaboration diagram for Exception:



## Public Member Functions

- [Exception](#) ()  
*Default constructor.*
- [Exception](#) (int line, const char \*fileName, const char \*funcName, const char \*buildDate, const char \*buildTime, const char \*errMsg, [Error](#) err)  
*Message constructor.*
- [Exception](#) (const [Exception](#) &except)  
*Copy constructor.*
- virtual [~Exception](#) () throw ()  
*Default destructor.*
- [Exception](#) & [operator=](#) (const [Exception](#) &except)  
*Assignment operator.*
- bool [operator==](#) (const [Error](#) err) const  
*Equality operator.*
- bool [operator!=](#) (const [Error](#) err) const  
*Inequality operator.*
- virtual const char \* [what](#) () const throw ()  
*virtual override for what().*
- const char \* [GetFullErrorMessage](#) () const  
*Gets the error code and full error message including the line, file, function, build date, and time.*
- const char \* [GetErrorMessage](#) () const  
*Accessor Functions.*
- const char \* [GetFileName](#) () const
- const char \* [GetFunctionName](#) () const
- const char \* [GetBuildDate](#) () const
- const char \* [GetBuildTime](#) () const
- int [GetLineNumber](#) () const
- [Error](#) [GetError](#) () const

### 10.69.1 Detailed Description

The [Exception](#) object represents an error that is returned from the library.

Overloaded operators allow comparisons against other [Exception](#) objects.

## 10.69.2 Constructor & Destructor Documentation

### 10.69.2.1 Exception ( )

Default constructor.

### 10.69.2.2 Exception ( int *line*, const char \* *fileName*, const char \* *funcName*, const char \* *buildDate*, const char \* *buildTime*, const char \* *errMsg*, Error *err* )

Message constructor.

Parameters

|                  |                                           |
|------------------|-------------------------------------------|
| <i>line</i>      | Line number where the exception is thrown |
| <i>fileName</i>  | Name of the file called                   |
| <i>funcName</i>  | Name of the function called               |
| <i>buildDate</i> | Build date                                |
| <i>buildTime</i> | Build time                                |
| <i>errMsg</i>    | A pointer to the exception message string |
| <i>err</i>       | Error code                                |

### 10.69.2.3 Exception ( const Exception & *except* )

Copy constructor.

### 10.69.2.4 virtual ~Exception ( ) throw ) [virtual]

Default destructor.

## 10.69.3 Member Function Documentation

### 10.69.3.1 const char\* GetBuildDate ( ) const

### 10.69.3.2 const char\* GetBuildTime ( ) const

### 10.69.3.3 Error GetError ( ) const

### 10.69.3.4 const char\* GetErrorMessage ( ) const

Accessor Functions.

### 10.69.3.5 const char\* GetFileName ( ) const

### 10.69.3.6 const char\* GetFullErrorMessage ( ) const

Gets the error code and full error message including the line, file, function, build date, and time.

10.69.3.7 `const char* GetFunctionName ( ) const`

10.69.3.8 `int GetLineNumber ( ) const`

10.69.3.9 `bool operator!= ( const Error err ) const`

Inequality operator.

10.69.3.10 `Exception& operator= ( const Exception & except )`

Assignment operator.

10.69.3.11 `bool operator== ( const Error err ) const`

Equality operator.

10.69.3.12 `virtual const char* what ( ) const throw )` [virtual]

virtual override for [what\(\)](#).

Gets the error code and error message.

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

- include/[Exception.h](#)

## 10.70 EYesNoClass Class Reference

Holds conversion methods for the standard namespace enumeration.

### Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, [EYesNo](#) \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, [EYesNo](#) \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring ToString](#) ([EYesNo](#) Value)  
*Converts a string to an int32\_t property.*

### 10.70.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

## 10.70.2 Member Function Documentation

10.70.2.1 `static bool FromString ( const GenICam::gcstring & ValueStr, EYesNo * pValue ) [static]`

Converts a string to enum value.

10.70.2.2 `static void ToString ( GenICam::gcstring & ValueStr, EYesNo * pValue ) [static]`

Converts a string to an int32\_t property.

10.70.2.3 `static GenICam::gcstring ToString ( EYesNo Value ) [static]`

Converts a string to an int32\_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

## 10.71 FileProtocolAdapter Class Reference

Adapter between the std::iostreambuf and the SFNC Features representing the device file system.

### Public Member Functions

- [FileProtocolAdapter](#) ()  
*Constructor.*
- virtual [~FileProtocolAdapter](#) ()
- bool [attach](#) (::Spinnaker::GenApi::INodeMap \*pInterface)  
*attach file protocol adapter to [NodeMap](#)*
- bool [openFile](#) (const char \*pFileName, std::ios\_base::openmode mode)  
*open a file on the device*
- bool [closeFile](#) (const char \*pFileName)  
*close a file on the device*
- std::streamsize [write](#) (const char \*buf, int64\_t offs, int64\_t len, const char \*pFileName)  
*writes data into a file.*
- std::streamsize [read](#) (char \*buf, int64\_t offs, std::streamsize len, const char \*pFileName)  
*read data from the device into a buffer*
- int64\_t [getBufSize](#) (const char \*pFileName, std::ios\_base::openmode mode)  
*fetch max FileAccessBuffer length for a file*
- bool [deleteFile](#) (const char \*pFileName)  
*Delete the content of the file.*

### 10.71.1 Detailed Description

Adapter between the std::iostreambuf and the SFNC Features representing the device file system.

The adapter assumes, that the features provide stdio file access compatible semantic

## 10.71.2 Constructor & Destructor Documentation

### 10.71.2.1 FileProtocolAdapter ( )

Constructor.

### 10.71.2.2 virtual ~FileProtocolAdapter ( ) [virtual]

## 10.71.3 Member Function Documentation

### 10.71.3.1 bool attach ( ::Spinnaker::GenApi::INodeMap \* *pInterface* )

attach file protocol adapter to [NodeMap](#)

Parameters

|                   |                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------|
| <i>pInterface</i> | <a href="#">NodeMap</a> of the device to which the <a href="#">FileProtocolAdapter</a> is attached |
|-------------------|----------------------------------------------------------------------------------------------------|

Returns

true if attach was successful, false if not

### 10.71.3.2 bool closeFile ( const char \* *pFileName* )

close a file on the device

Parameters

|                  |                                                                                       |
|------------------|---------------------------------------------------------------------------------------|
| <i>pFileName</i> | filename of the file to open. The filename must exist in the Enumeration FileSelector |
|------------------|---------------------------------------------------------------------------------------|

Returns

true on success, false on error

### 10.71.3.3 bool deleteFile ( const char \* *pFileName* )

Delete the content of the file.

Parameters

|                  |                                                                                       |
|------------------|---------------------------------------------------------------------------------------|
| <i>pFileName</i> | filename of the file to open. The filename must exist in the Enumeration FileSelector |
|------------------|---------------------------------------------------------------------------------------|

**Returns**

true on success, false on error

**10.71.3.4** `int64_t getBufSize ( const char * pFileName, std::ios_base::openmode mode )`

fetch max FileAccessBuffer length for a file

**Parameters**

|                  |                                                                                       |
|------------------|---------------------------------------------------------------------------------------|
| <i>pFileName</i> | filename of the file to open. The filename must exist in the Enumeration FileSelector |
| <i>mode</i>      | mode to open the file. The mode must exist in the Enumeration FileOpenMode            |

**Returns**

max length of FileAccessBuffer in the given mode on the given file

**10.71.3.5** `bool openFile ( const char * pFileName, std::ios_base::openmode mode )`

open a file on the device

**Parameters**

|                  |                                                                                       |
|------------------|---------------------------------------------------------------------------------------|
| <i>pFileName</i> | filename of the file to open. The filename must exist in the Enumeration FileSelector |
| <i>mode</i>      | mode to open the file. The mode must exist in the Enumeration FileOpenMode            |

**Returns**

true on success, false on error

**10.71.3.6** `std::streamsize read ( char * buf, int64_t offs, std::streamsize len, const char * pFileName )`

read data from the device into a buffer

**Parameters**

|                  |                                                                                            |
|------------------|--------------------------------------------------------------------------------------------|
| <i>buf</i>       | target buffer                                                                              |
| <i>offs</i>      | offset in the device file to read from                                                     |
| <i>len</i>       | count of bytes to read                                                                     |
| <i>pFileName</i> | filename of the file to write into The filename must exist in the Enumeration FileSelector |

**Returns**

count of bytes successfully read



10.71.3.7 `std::streamsize write ( const char * buf, int64_t offs, int64_t len, const char * pFileName )`

writes data into a file.

#### Parameters

|                  |                                                                                            |
|------------------|--------------------------------------------------------------------------------------------|
| <i>buf</i>       | source buffer                                                                              |
| <i>offs</i>      | offset into the device file                                                                |
| <i>len</i>       | count of bytes to write                                                                    |
| <i>pFileName</i> | filename of the file to write into The filename must exist in the Enumeration FileSelector |

#### Returns

count of bytes written

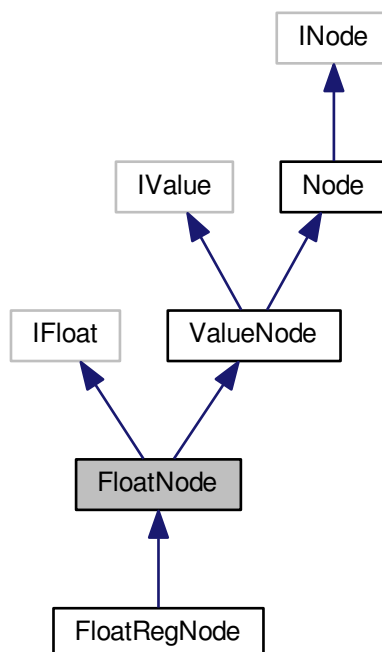
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Filestream.h](#)

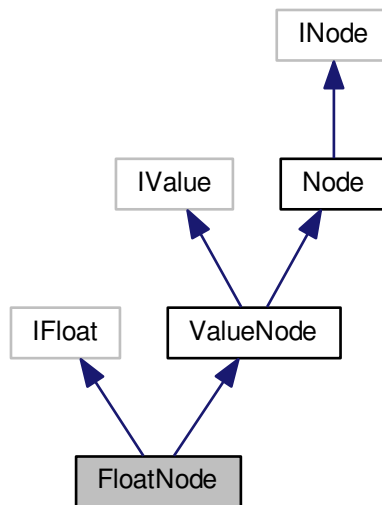
## 10.72 FloatNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for FloatNode:



Collaboration diagram for FloatNode:



## Public Member Functions

- [FloatNode](#) ()
- [FloatNode](#) (std::shared\_ptr< Node::NodeImpl > pFloat)
- virtual [~FloatNode](#) ()
- virtual void [SetValue](#) (double Value, bool [Verify](#)=true)
  - Set node value.*
- virtual [IFloat](#) & [operator=](#) (double Value)
  - Set node value.*
- virtual double [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
  - Get node value.*
- virtual double [operator\(\)](#) ()
  - Get node value.*
- virtual double [operator\\*](#) ()
  - Get node value.*
- virtual double [GetMin](#) ()
  - Get minimum value allowed.*
- virtual double [GetMax](#) ()
  - Get maximum value allowed.*
- virtual bool [HasInc](#) ()
  - True if the float has a constant increment.*
- virtual [EIncMode](#) [GetIncMode](#) ()
  - Get increment mode.*
- virtual double [GetInc](#) ()
  - Get the constant increment if there is any.*
- virtual [double\\_autovector\\_t](#) [GetListOfValidValues](#) (bool bounded=true)
  - Get list of valid value.*

- virtual [ERepresentation](#) [GetRepresentation](#) ()  
*Get recommended representation.*
- virtual [GenICam::gcstring](#) [GetUnit](#) () const  
*Get the physical unit name.*
- virtual [EDisplayNotation](#) [GetDisplayNotation](#) () const  
*Get the way the float should be converted to a string.*
- virtual [int64\\_t](#) [GetDisplayPrecision](#) () const  
*Get the precision to be used when converting the float to a string.*
- [Integer](#) \* [GetIntAlias](#) ()  
*gets the interface of an alias node.*
- [IEnumeration](#) \* [GetEnumAlias](#) ()  
*gets the interface of an alias node.*
- virtual void [ImposeMin](#) (double Value)  
*Restrict minimum value.*
- virtual void [ImposeMax](#) (double Value)  
*Restrict maximum value.*
- virtual void [SetReference](#) ([INode](#) \*pBase)  
*overload SetReference for Float*

## Additional Inherited Members

### 10.72.1 Detailed Description

[Interface](#) for string properties.

### 10.72.2 Constructor & Destructor Documentation

#### 10.72.2.1 [FloatNode](#) ( )

#### 10.72.2.2 [FloatNode](#) ( [std::shared\\_ptr](#)< [Node::NodeImpl](#) > pFloat )

#### 10.72.2.3 virtual [~FloatNode](#) ( ) [virtual]

### 10.72.3 Member Function Documentation

#### 10.72.3.1 virtual [EDisplayNotation](#) [GetDisplayNotation](#) ( ) const [virtual]

Get the way the float should be converted to a string.

#### 10.72.3.2 virtual [int64\\_t](#) [GetDisplayPrecision](#) ( ) const [virtual]

Get the precision to be used when converting the float to a string.

#### 10.72.3.3 [IEnumeration](#)\* [GetEnumAlias](#) ( )

gets the interface of an alias node.

10.72.3.4 `virtual double GetInc ( ) [virtual]`

Get the constant increment if there is any.

10.72.3.5 `virtual EIncMode GetIncMode ( ) [virtual]`

Get increment mode.

10.72.3.6 `Integer* GetIntAlias ( )`

gets the interface of an alias node.

10.72.3.7 `virtual double_autovector_t GetListOfValidValues ( bool bounded = true ) [virtual]`

Get list of valid value.

10.72.3.8 `virtual double GetMax ( ) [virtual]`

Get maximum value allowed.

10.72.3.9 `virtual double GetMin ( ) [virtual]`

Get minimum value allowed.

10.72.3.10 `virtual ERepresentation GetRepresentation ( ) [virtual]`

Get recommended representation.

10.72.3.11 `virtual GenICam::gcstring GetUnit ( ) const [virtual]`

Get the physical unit name.

10.72.3.12 `virtual double GetValue ( bool Verify = false, bool IgnoreCache = false ) [virtual]`

Get node value.

Parameters

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| <i>Verify</i>      | Enables Range verification (default = false). The AccessMode is always checked |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false)                |

**Returns**

The value read

**10.72.3.13** `virtual bool HasInc ( )` [virtual]

True if the float has a constant increment.

**10.72.3.14** `virtual void ImposeMax ( double Value )` [virtual]

Restrict maximum value.

**10.72.3.15** `virtual void ImposeMin ( double Value )` [virtual]

Restrict minimum value.

**10.72.3.16** `virtual double operator()( )` [virtual]

Get node value.

**10.72.3.17** `virtual double operator*( )` [virtual]

Get node value.

**10.72.3.18** `virtual IFloat& operator=( double Value )` [virtual]

Set node value.

**10.72.3.19** `virtual void SetReference ( INode * pBase )` [virtual]

overload SetReference for Float

Reimplemented from [ValueNode](#).

Reimplemented in [FloatRegNode](#).

**10.72.3.20** `virtual void SetValue ( double Value, bool Verify = true )` [virtual]

Set node value.

**Parameters**

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>Value</i>  | The value to set                                           |
| <i>Verify</i> | Enables AccessMode and Range verification (default = true) |

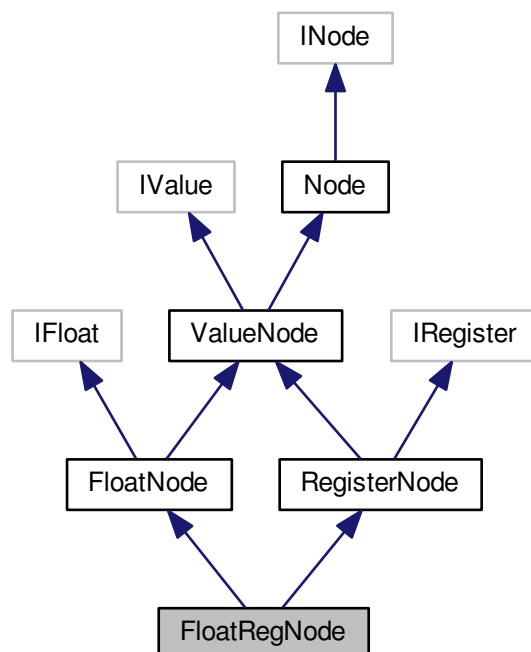
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatNode.h](#)

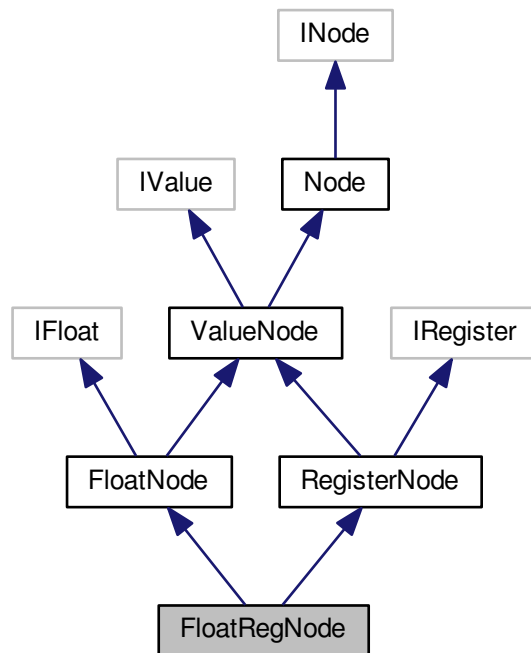
## 10.73 FloatRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for FloatRegNode:



Collaboration diagram for FloatRegNode:



## Public Member Functions

- [FloatRegNode](#) ()
- [FloatRegNode](#) (std::shared\_ptr< Node::NodeImpl > pFloat)
- virtual [~FloatRegNode](#) ()
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.73.1 Detailed Description

[Interface](#) for string properties.

### 10.73.2 Constructor & Destructor Documentation

#### 10.73.2.1 FloatRegNode ( )

#### 10.73.2.2 FloatRegNode ( std::shared\_ptr< Node::NodeImpl > pFloat )

10.73.2.3 `virtual ~FloatRegNode ( ) [virtual]`

### 10.73.3 Member Function Documentation

10.73.3.1 `virtual void SetReference ( INode * pBase ) [virtual]`

overload SetReference for Value

Reimplemented from [FloatNode](#).

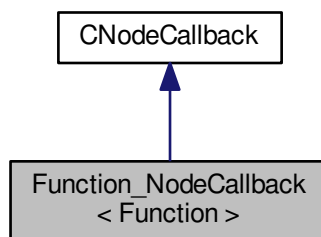
The documentation for this class was generated from the following file:

- `include/SpinGenApi/FloatRegNode.h`

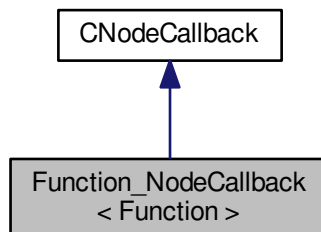
## 10.74 `Function_NodeCallback< Function >` Class Template Reference

Container for a function pointer.

Inheritance diagram for `Function_NodeCallback< Function >`:



Collaboration diagram for `Function_NodeCallback< Function >`:





## Public Member Functions

- [Function\\_NodeCallback](#) (INode \*pNode, const Function &function, [ECallbackType](#) CallbackType)  
*Constructor.*
- virtual void [operator\(\)](#) ([ECallbackType](#) CallbackType) const  
*execute operation: call the function*
- virtual void [Destroy](#) ()  
*destroys teh object*

## Additional Inherited Members

### 10.74.1 Detailed Description

```
template<class Function>
class Spinnaker::GenApi::Function_NodeCallback< Function >
```

Container for a function pointer.

### 10.74.2 Constructor & Destructor Documentation

10.74.2.1 [Function\\_NodeCallback](#) ( INode \* pNode, const Function & function, [ECallbackType](#) CallbackType )  
[inline]

Constructor.

### 10.74.3 Member Function Documentation

10.74.3.1 virtual void [Destroy](#) ( ) [inline],[virtual]

destroys teh object

Implements [CNodeCallback](#).

10.74.3.2 virtual void [operator\(\)](#) ( [ECallbackType](#) CallbackType ) const [inline],[virtual]

execute operation: call the function

Implements [CNodeCallback](#).

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

- include/SpinGenApi/[NodeCallback.h](#)

## 10.75 gcstring Class Reference

### Public Member Functions

- [gcstring](#) ()
- [gcstring](#) (const char \*pc)
- [gcstring](#) (const char \*pc, size\_t n)
- [gcstring](#) (size\_t count, char ch)
- [gcstring](#) (const [gcstring](#) &str)
- virtual [~gcstring](#) (void)
- virtual [gcstring](#) & [append](#) (const [gcstring](#) &str)
- virtual [gcstring](#) & [append](#) (size\_t count, char ch)
- virtual [gcstring](#) & [assign](#) (const [gcstring](#) &str)
- virtual [gcstring](#) & [assign](#) (size\_t count, char ch)
- virtual [gcstring](#) & [assign](#) (const char \*pc)
- virtual [gcstring](#) & [assign](#) (const char \*pc, size\_t n)
- virtual int [compare](#) (const [gcstring](#) &str) const
- virtual const char \* [c\\_str](#) (void) const
- virtual bool [empty](#) (void) const
- virtual size\_t [find](#) (char ch, size\_t offset=0) const
- virtual size\_t [find](#) (const [gcstring](#) &str, size\_t offset=0) const
- virtual size\_t [find](#) (const [gcstring](#) &str, size\_t offset, size\_t count) const
- virtual size\_t [find](#) (const char \*pc, size\_t offset=0) const
- virtual size\_t [find](#) (const char \*pc, size\_t offset, size\_t count) const
- virtual size\_t [length](#) (void) const
- virtual size\_t [size](#) (void) const
- virtual void [resize](#) (size\_t n)
- virtual size\_t [max\\_size](#) () const
- virtual [gcstring](#) [substr](#) (size\_t offset=0, size\_t count=GCSTRING\_NPOS) const
- virtual size\_t [find\\_first\\_of](#) (const [gcstring](#) &str, size\_t offset=0) const
- virtual size\_t [find\\_first\\_not\\_of](#) (const [gcstring](#) &str, size\_t offset=0) const
- virtual void [swap](#) ([gcstring](#) &Right)
- bool [operator!=](#) (const [gcstring](#) &str) const
- bool [operator!=](#) (const char \*pc) const
- [gcstring](#) & [operator+=](#) (const [gcstring](#) &str)
- [gcstring](#) [operator+=](#) (const [gcstring](#) &str) const
- [gcstring](#) & [operator+=](#) (const char \*pc)
- [gcstring](#) & [operator+=](#) (char ch)
- [gcstring](#) [operator+=](#) (char ch) const
- [gcstring](#) & [operator=](#) (const [gcstring](#) &str)
- bool [operator==](#) (const [gcstring](#) &str) const
- bool [operator==](#) (const char \*pc) const
- bool [operator<](#) (const [gcstring](#) &str) const
- bool [operator>](#) (const [gcstring](#) &str) const
- [operator](#) const char \* (void) const
- void [operator delete](#) (void \*pWhere)
- void [operator delete](#) (void \*pWhere, void \*pNewWhere)
- void \* [operator new](#) (size\_t uiSize)
- void \* [operator new](#) (size\_t uiSize, void \*pWhere)

### Static Public Member Functions

- static size\_t [\\_npos](#) (void)

## Static Public Attributes

- static const size\_t [npos](#)

## Friends

- [SPINNAKER\\_API](#) friend [gcstring operator+](#) (const [gcstring](#) &left, const [gcstring](#) &right)
- [SPINNAKER\\_API](#) friend [gcstring operator+](#) (const [gcstring](#) &left, const char \*right)
- [SPINNAKER\\_API](#) friend [gcstring operator+](#) (const char \*left, const [gcstring](#) &right)

## 10.75.1 Constructor & Destructor Documentation

10.75.1.1 [gcstring](#) ( )

10.75.1.2 [gcstring](#) ( const char \* *pc* )

10.75.1.3 [gcstring](#) ( const char \* *pc*, size\_t *n* )

10.75.1.4 [gcstring](#) ( size\_t *count*, char *ch* )

10.75.1.5 [gcstring](#) ( const [gcstring](#) & *str* )

10.75.1.6 virtual ~[gcstring](#) ( void ) [virtual]

## 10.75.2 Member Function Documentation

10.75.2.1 static size\_t [npos](#) ( void ) [static]

10.75.2.2 virtual [gcstring&](#) append ( const [gcstring](#) & *str* ) [virtual]

10.75.2.3 virtual [gcstring&](#) append ( size\_t *count*, char *ch* ) [virtual]

10.75.2.4 virtual [gcstring&](#) assign ( const [gcstring](#) & *str* ) [virtual]

10.75.2.5 virtual [gcstring&](#) assign ( size\_t *count*, char *ch* ) [virtual]

10.75.2.6 virtual [gcstring&](#) assign ( const char \* *pc* ) [virtual]

10.75.2.7 virtual [gcstring&](#) assign ( const char \* *pc*, size\_t *n* ) [virtual]

10.75.2.8 virtual const char\* [c\\_str](#) ( void ) const [virtual]

10.75.2.9 virtual int compare ( const [gcstring](#) & *str* ) const [virtual]

10.75.2.10 virtual bool empty ( void ) const [virtual]

- 10.75.2.11 virtual size\_t find ( char *ch*, size\_t *offset* = 0 ) const [virtual]
- 10.75.2.12 virtual size\_t find ( const gcstring & *str*, size\_t *offset* = 0 ) const [virtual]
- 10.75.2.13 virtual size\_t find ( const gcstring & *str*, size\_t *offset*, size\_t *count* ) const [virtual]
- 10.75.2.14 virtual size\_t find ( const char \* *pc*, size\_t *offset* = 0 ) const [virtual]
- 10.75.2.15 virtual size\_t find ( const char \* *pc*, size\_t *offset*, size\_t *count* ) const [virtual]
- 10.75.2.16 virtual size\_t find\_first\_not\_of ( const gcstring & *str*, size\_t *offset* = 0 ) const [virtual]
- 10.75.2.17 virtual size\_t find\_first\_of ( const gcstring & *str*, size\_t *offset* = 0 ) const [virtual]
- 10.75.2.18 virtual size\_t length ( void ) const [virtual]
- 10.75.2.19 virtual size\_t max\_size ( ) const [virtual]
- 10.75.2.20 operator const char \* ( void ) const
- 10.75.2.21 void operator delete ( void \* *pWhere* )
- 10.75.2.22 void operator delete ( void \* *pWhere*, void \* *pNewWhere* )
- 10.75.2.23 void\* operator new ( size\_t *uiSize* )
- 10.75.2.24 void\* operator new ( size\_t *uiSize*, void \* *pWhere* )
- 10.75.2.25 bool operator!= ( const gcstring & *str* ) const
- 10.75.2.26 bool operator!= ( const char \* *pc* ) const
- 10.75.2.27 gcstring& operator+= ( const gcstring & *str* )
- 10.75.2.28 gcstring operator+= ( const gcstring & *str* ) const
- 10.75.2.29 gcstring& operator+= ( const char \* *pc* )
- 10.75.2.30 gcstring& operator+= ( char *ch* )
- 10.75.2.31 gcstring operator+= ( char *ch* ) const
- 10.75.2.32 bool operator< ( const gcstring & *str* ) const
- 10.75.2.33 gcstring& operator= ( const gcstring & *str* )

10.75.2.34 `bool operator==( const gcstring & str ) const`

10.75.2.35 `bool operator==( const char * pc ) const`

10.75.2.36 `bool operator>( const gcstring & str ) const`

10.75.2.37 `virtual void resize( size_t n ) [virtual]`

10.75.2.38 `virtual size_t size( void ) const [virtual]`

10.75.2.39 `virtual gcstring substr( size_t offset = 0, size_t count = GCSTRING_NPOS ) const [virtual]`

10.75.2.40 `virtual void swap( gcstring & Right ) [virtual]`

### 10.75.3 Friends And Related Function Documentation

10.75.3.1 `SPINNAKER_API friend gcstring operator+( const gcstring & left, const gcstring & right ) [friend]`

10.75.3.2 `SPINNAKER_API friend gcstring operator+( const gcstring & left, const char * right ) [friend]`

10.75.3.3 `SPINNAKER_API friend gcstring operator+( const char * left, const gcstring & right ) [friend]`

### 10.75.4 Member Data Documentation

10.75.4.1 `const size_t npos [static]`

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

- `include/SpinGenApi/GCString.h`

## 10.76 GVCP\_CHUNK\_TRAILER Struct Reference

header of a GVCP request packet

### Public Attributes

- `uint32_t ChunkID`
- `uint32_t ChunkLength`

### 10.76.1 Detailed Description

header of a GVCP request packet

## 10.76.2 Member Data Documentation

10.76.2.1 `uint32_t` ChunkID

10.76.2.2 `uint32_t` ChunkLength

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

- `include/SpinGenApi/ChunkAdapterGEV.h`

## 10.77 GVCP\_EVENT\_ITEM Struct Reference

layout of a GVCP event item (Extended ID flag not set)

### Public Attributes

- `uint16_t` [ReservedOrEventSize](#)
- `uint16_t` [EventId](#)
- `uint16_t` [StreamChannelId](#)
- `uint16_t` [BlockId](#)
- `uint32_t` [TimestampHigh](#)
- `uint32_t` [TimestampLow](#)

### 10.77.1 Detailed Description

layout of a GVCP event item (Extended ID flag not set)

## 10.77.2 Member Data Documentation

10.77.2.1 `uint16_t` BlockId

10.77.2.2 `uint16_t` EventId

10.77.2.3 `uint16_t` ReservedOrEventSize

10.77.2.4 `uint16_t` StreamChannelId

10.77.2.5 `uint32_t` TimestampHigh

10.77.2.6 `uint32_t` TimestampLow

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

- `include/SpinGenApi/EventAdapterGEV.h`

## 10.78 GVCP\_EVENT\_ITEM\_BASIC Struct Reference

layout of a GVCP event item (common to all types)

### Public Attributes

- uint16\_t [ReservedOrEventSize](#)
- uint16\_t [EventId](#)

### 10.78.1 Detailed Description

layout of a GVCP event item (common to all types)

### 10.78.2 Member Data Documentation

10.78.2.1 uint16\_t EventId

10.78.2.2 uint16\_t ReservedOrEventSize

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 10.79 GVCP\_EVENT\_ITEM\_EXTENDED\_ID Struct Reference

layout of a GVCP event item (Extended ID flag set)

### Public Attributes

- uint16\_t [ReservedOrEventSize](#)
- uint16\_t [EventId](#)
- uint16\_t [StreamChannelId](#)
- uint16\_t [BlockId](#)
- uint32\_t [BlockId64High](#)
- uint32\_t [BlockId64Low](#)
- uint32\_t [TimestampHigh](#)
- uint32\_t [TimestampLow](#)

### 10.79.1 Detailed Description

layout of a GVCP event item (Extended ID flag set)

## 10.79.2 Member Data Documentation

10.79.2.1 `uint16_t` BlockId

10.79.2.2 `uint32_t` BlockId64High

10.79.2.3 `uint32_t` BlockId64Low

10.79.2.4 `uint16_t` EventId

10.79.2.5 `uint16_t` ReservedOrEventSize

10.79.2.6 `uint16_t` StreamChannelId

10.79.2.7 `uint32_t` TimestampHigh

10.79.2.8 `uint32_t` TimestampLow

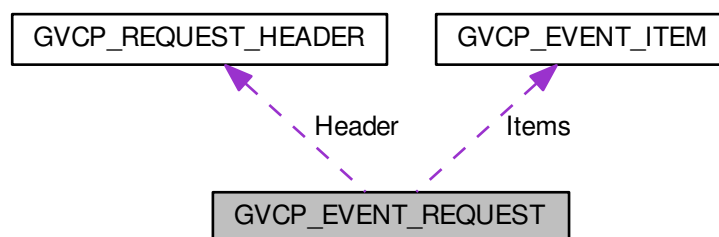
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterGEV.h`

## 10.80 GVCP\_EVENT\_REQUEST Struct Reference

Layout of a GVCP event request packet (Extended ID flag not set)

Collaboration diagram for GVCP\_EVENT\_REQUEST:



### Public Attributes

- `GVCP_REQUEST_HEADER` Header
- `GVCP_EVENT_ITEM` Items [1]



### 10.80.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag not set)

### 10.80.2 Member Data Documentation

#### 10.80.2.1 GVCP\_REQUEST\_HEADER Header

#### 10.80.2.2 GVCP\_EVENT\_ITEM Items[1]

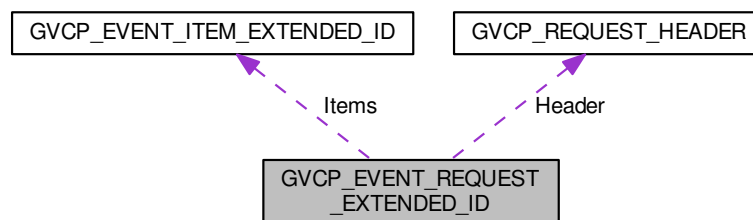
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 10.81 GVCP\_EVENT\_REQUEST\_EXTENDED\_ID Struct Reference

Layout of a GVCP event request packet (Extended ID flag set)

Collaboration diagram for GVCP\_EVENT\_REQUEST\_EXTENDED\_ID:



### Public Attributes

- [GVCP\\_REQUEST\\_HEADER](#) Header
- [GVCP\\_EVENT\\_ITEM\\_EXTENDED\\_ID](#) Items [1]

### 10.81.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag set)

### 10.81.2 Member Data Documentation

#### 10.81.2.1 GVCP\_REQUEST\_HEADER Header

#### 10.81.2.2 GVCP\_EVENT\_ITEM\_EXTENDED\_ID Items[1]

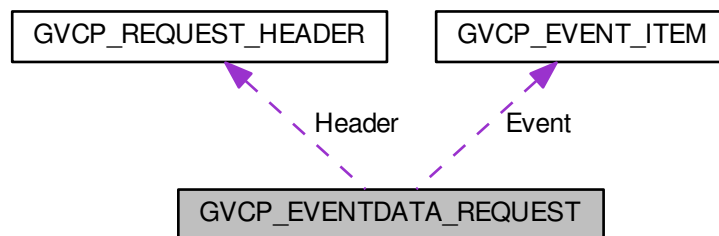
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 10.82 GVCP\_EVENTDATA\_REQUEST Struct Reference

Layout of a GVCP event data request packet (Extended ID flag not set)

Collaboration diagram for GVCP\_EVENTDATA\_REQUEST:



### Public Attributes

- [GVCP\\_REQUEST\\_HEADER](#) Header
- [GVCP\\_EVENT\\_ITEM](#) Event
- uint32\_t [Data](#) [1]

### 10.82.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag not set)

### 10.82.2 Member Data Documentation

#### 10.82.2.1 uint32\_t Data[1]

#### 10.82.2.2 GVCP\_EVENT\_ITEM Event

#### 10.82.2.3 GVCP\_REQUEST\_HEADER Header

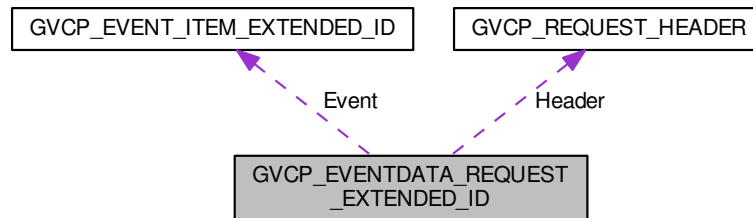
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 10.83 GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID Struct Reference

Layout of a GVCP event data request packet (Extended ID flag set)

Collaboration diagram for GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID:



### Public Attributes

- [GVCP\\_REQUEST\\_HEADER](#) Header
- [GVCP\\_EVENT\\_ITEM\\_EXTENDED\\_ID](#) Event
- [uint32\\_t](#) [Data](#) [1]

### 10.83.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag set)

### 10.83.2 Member Data Documentation

10.83.2.1 [uint32\\_t](#) [Data](#)[1]

10.83.2.2 [GVCP\\_EVENT\\_ITEM\\_EXTENDED\\_ID](#) Event

10.83.2.3 [GVCP\\_REQUEST\\_HEADER](#) Header

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

- `include/SpinGenApi/EventAdapterGEV.h`

## 10.84 GVCP\_REQUEST\_HEADER Struct Reference

header of a GVCP request packet

## Public Attributes

- uint8\_t [Magic](#)
- uint8\_t [Flags](#)
- uint16\_t [Command](#)
- uint16\_t [Length](#)
- uint16\_t [ReqId](#)

### 10.84.1 Detailed Description

header of a GVCP request packet

### 10.84.2 Member Data Documentation

10.84.2.1 [uint16\\_t Command](#)

10.84.2.2 [uint8\\_t Flags](#)

10.84.2.3 [uint16\\_t Length](#)

10.84.2.4 [uint8\\_t Magic](#)

10.84.2.5 [uint16\\_t ReqId](#)

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 10.85 H264Option Struct Reference

Options for saving H264 files.

### Public Member Functions

- [H264Option](#) ()

### Public Attributes

- float [frameRate](#)  
*Frame rate of the stream.*
- unsigned int [width](#)  
*Width of source image.*
- unsigned int [height](#)  
*Height of source image.*
- unsigned int [bitrate](#)  
*Bit-rate to encode at.*
- unsigned int [reserved](#) [256]  
*Reserved for future use.*

### 10.85.1 Detailed Description

Options for saving H264 files.

### 10.85.2 Constructor & Destructor Documentation

#### 10.85.2.1 H264Option ( ) `[inline]`

### 10.85.3 Member Data Documentation

#### 10.85.3.1 unsigned int bitrate

Bit-rate to encode at.

#### 10.85.3.2 float frameRate

Frame rate of the stream.

#### 10.85.3.3 unsigned int height

Height of source image.

#### 10.85.3.4 unsigned int reserved[256]

Reserved for future use.

#### 10.85.3.5 unsigned int width

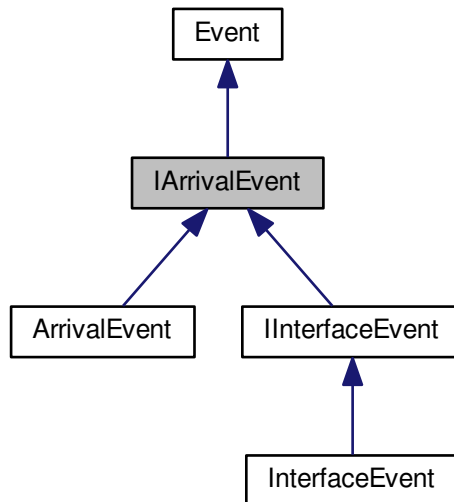
Width of source image.

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

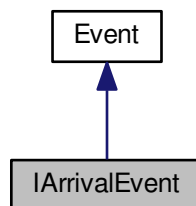
- [include/SpinVideoDefs.h](#)

## 10.86 IArrivalEvent Class Reference

Inheritance diagram for IArrivalEvent:



Collaboration diagram for IArrivalEvent:



### Public Member Functions

- virtual [~IArrivalEvent](#) ()
- virtual void [OnDeviceArrival](#) (uint64\_t serialNumber)=0

### Protected Member Functions

- [IArrivalEvent](#) ()
- [IArrivalEvent](#) (const [IArrivalEvent](#) &)
- [IArrivalEvent](#) & [operator=](#) (const [IArrivalEvent](#) &)

## Additional Inherited Members

### 10.86.1 Constructor & Destructor Documentation

10.86.1.1 `virtual ~IArrivalEvent ( ) [inline],[virtual]`

10.86.1.2 `IArrivalEvent ( ) [inline],[protected]`

10.86.1.3 `IArrivalEvent ( const IArrivalEvent & ) [inline],[protected]`

### 10.86.2 Member Function Documentation

10.86.2.1 `virtual void OnDeviceArrival ( uint64_t serialNumber ) [pure virtual]`

Implemented in [ArrivalEvent](#), [InterfaceEvent](#), and [IInterfaceEvent](#).

10.86.2.2 `IArrivalEvent& operator= ( const IArrivalEvent & ) [protected]`

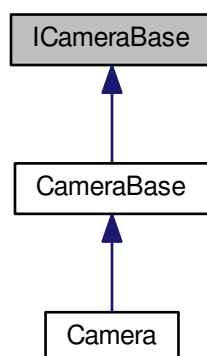
The documentation for this class was generated from the following file:

- `include/Interface/IArrivalEvent.h`

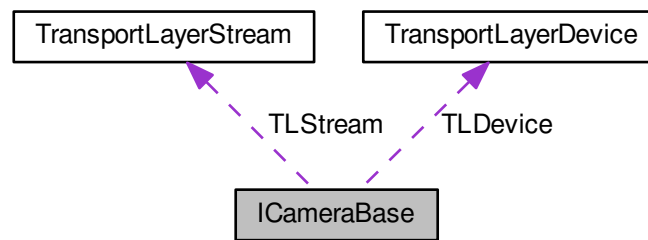
## 10.87 ICameraBase Class Reference

The interface file for base class for the camera object.

Inheritance diagram for ICameraBase:



Collaboration diagram for ICameraBase:



## Public Member Functions

- virtual `~ICameraBase` (void)
- virtual void `Init` ()=0
- virtual void `Delnit` ()=0
- virtual bool `IsInitialized` ()=0
- virtual bool `IsValid` ()=0
- virtual `GenApi::INodeMap` & `GetNodeMap` () const =0
- virtual `GenApi::INodeMap` & `GetTLDeviceNodeMap` () const =0
- virtual `GenApi::INodeMap` & `GetTLStreamNodeMap` () const =0
- virtual `GenApi::EAccessMode` `GetAccessMode` () const =0
- virtual void `ReadPort` (uint64\_t iAddress, void \*pBuffer, size\_t iSize)=0
- virtual void `WritePort` (uint64\_t iAddress, const void \*pBuffer, size\_t iSize)=0
- virtual void `BeginAcquisition` ()=0
- virtual void `EndAcquisition` ()=0
- virtual `ImagePtr` `GetNextImage` (uint64\_t grabTimeout=EVENT\_TIMEOUT\_INFINITE, uint64\_t streamID=0)=0
- virtual `GenICam::gcstring` `GetUniqueID` ()=0
- virtual bool `IsStreaming` () const =0
- virtual `GenICam::gcstring` `GetGuiXml` () const =0
- virtual void `RegisterEvent` (`Event` &evtToRegister)=0
- virtual void `RegisterEvent` (`Event` &evtToRegister, const `GenICam::gcstring` &eventName)=0
- virtual void `UnregisterEvent` (`Event` &evtToUnregister)=0
- virtual unsigned int `GetNumImagesInUse` ()=0
- virtual unsigned int `GetNumDataStreams` ()=0
- virtual unsigned int `DiscoverMaxPacketSize` ()=0

## Public Attributes

- `TransportLayerDevice` `TLDevice`  
*Gets vital camera information by connecting to the camera's bootstrap registers.*
- `TransportLayerStream` `TLStream`  
*Gets information about the stream data by connecting to the camera's bootstrap registers.*



## Protected Member Functions

- [ICameraBase](#) ()
- [ICameraBase](#) (const [ICameraBase](#) &)
- [ICameraBase](#) & [operator=](#) (const [ICameraBase](#) &)

## Protected Attributes

- CameraBaseData \* [m\\_pCameraBaseData](#)

## Friends

- class [CameraInternal](#)
- class [InterfaceImpl](#)

### 10.87.1 Detailed Description

The interface file for base class for the camera object.

### 10.87.2 Constructor & Destructor Documentation

10.87.2.1 `virtual ~ICameraBase ( void ) [inline],[virtual]`

10.87.2.2 `ICameraBase ( ) [inline],[protected]`

10.87.2.3 `ICameraBase ( const ICameraBase & ) [inline],[protected]`

### 10.87.3 Member Function Documentation

10.87.3.1 `virtual void BeginAcquisition ( ) [pure virtual]`

Implemented in [CameraBase](#).

10.87.3.2 `virtual void Delnit ( ) [pure virtual]`

Implemented in [CameraBase](#).

10.87.3.3 `virtual unsigned int DiscoverMaxPacketSize ( ) [pure virtual]`

Implemented in [CameraBase](#).

10.87.3.4 `virtual void EndAcquisition ( ) [pure virtual]`

Implemented in [CameraBase](#).

10.87.3.5 virtual **GenApi::EAccessMode** GetAccessMode ( ) const [pure virtual]

Implemented in [CameraBase](#).

10.87.3.6 virtual **GenICam::gcstring** GetGuiXml ( ) const [pure virtual]

Implemented in [CameraBase](#).

10.87.3.7 virtual **ImagePtr** GetNextImage ( uint64\_t *grabTimeout* = **EVENT\_TIMEOUT\_INFINITE**, uint64\_t *streamID* = 0 )  
[pure virtual]

Implemented in [CameraBase](#).

10.87.3.8 virtual **GenApi::INodeMap&** GetNodeMap ( ) const [pure virtual]

Implemented in [CameraBase](#).

10.87.3.9 virtual unsigned int GetNumDataStreams ( ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.10 virtual unsigned int GetNumImagesInUse ( ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.11 virtual **GenApi::INodeMap&** GetTLDeviceNodeMap ( ) const [pure virtual]

Implemented in [CameraBase](#).

10.87.3.12 virtual **GenApi::INodeMap&** GetTLStreamNodeMap ( ) const [pure virtual]

Implemented in [CameraBase](#).

10.87.3.13 virtual **GenICam::gcstring** GetUniqueID ( ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.14 virtual void Init ( ) [pure virtual]

Implemented in [CameraBase](#), and [Camera](#).

10.87.3.15 virtual bool IsInitialized ( ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.16 virtual bool IsStreaming ( ) const [pure virtual]

Implemented in [CameraBase](#).

10.87.3.17 virtual bool IsValid ( ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.18 ICameraBase& operator= ( const ICameraBase & ) [protected]

10.87.3.19 virtual void ReadPort ( uint64\_t iAddress, void \* pBuffer, size\_t iSize ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.20 virtual void RegisterEvent ( Event & evtToRegister ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.21 virtual void RegisterEvent ( Event & evtToRegister, const GenICam::gcstring & eventName ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.22 virtual void UnregisterEvent ( Event & evtToUnregister ) [pure virtual]

Implemented in [CameraBase](#).

10.87.3.23 virtual void WritePort ( uint64\_t iAddress, const void \* pBuffer, size\_t iSize ) [pure virtual]

Implemented in [CameraBase](#).

### 10.87.4 Friends And Related Function Documentation

10.87.4.1 friend class CameraInternal [friend]

10.87.4.2 friend class InterfaceImpl [friend]

### 10.87.5 Member Data Documentation

10.87.5.1 CameraBaseData\* m\_pCameraBaseData [protected]

#### 10.87.5.2 TransportLayerDevice TLDevice

Gets vital camera information by connecting to the camera's bootstrap registers.

These nodes also access host software modules and the nodes can be used without having to call [Init\(\)](#) on the camera.

#### 10.87.5.3 TransportLayerStream TLStream

Gets information about the stream data by connecting to the camera's bootstrap registers.

These nodes also access host software modules and the nodes can be used without having to call [Init\(\)](#) on the camera.

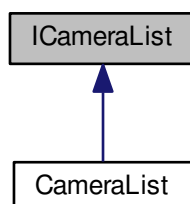
The documentation for this class was generated from the following file:

- include/Interface/[ICameraBase.h](#)

## 10.88 ICameraList Class Reference

Used to hold a list of camera objects.

Inheritance diagram for ICameraList:



## Public Member Functions

- virtual [~ICameraList](#) ()
- virtual [CameraPtr operator\[\]](#) (unsigned int index)=0
- virtual unsigned int [GetSize](#) () const =0
- virtual [CameraPtr GetByIndex](#) (unsigned int index) const =0
- virtual [CameraPtr GetBySerial](#) (std::string serialNumber) const =0
- virtual void [Clear](#) ()=0
- virtual void [RemoveBySerial](#) (std::string serialNumber)=0
- virtual void [RemoveByIndex](#) (unsigned int index)=0
- virtual void [Append](#) ([CameraList](#) &otherList)=0

## Protected Member Functions

- [ICameraList](#) ()
- [ICameraList](#) (const [ICameraList](#) &)
- [ICameraList](#) & [operator=](#) (const [ICameraList](#) &)

## Protected Attributes

- CameraListData \* [m\\_pCameraListData](#)

## Friends

- class [InterfaceImpl](#)
- class [CameraListImpl](#)

### 10.88.1 Detailed Description

Used to hold a list of camera objects.

### 10.88.2 Constructor & Destructor Documentation

10.88.2.1 virtual [~ICameraList](#) ( ) [inline],[virtual]

10.88.2.2 [ICameraList](#) ( ) [inline],[protected]

10.88.2.3 [ICameraList](#) ( const [ICameraList](#) & ) [inline],[protected]

### 10.88.3 Member Function Documentation

10.88.3.1 virtual void [Append](#) ( [CameraList](#) & *otherList* ) [pure virtual]

Implemented in [CameraList](#).

10.88.3.2 virtual void Clear ( ) [pure virtual]

Implemented in [CameraList](#).

10.88.3.3 virtual CameraPtr GetByIndex ( unsigned int *index* ) const [pure virtual]

Implemented in [CameraList](#).

10.88.3.4 virtual CameraPtr GetBySerial ( std::string *serialNumber* ) const [pure virtual]

Implemented in [CameraList](#).

10.88.3.5 virtual unsigned int GetSize ( ) const [pure virtual]

Implemented in [CameraList](#).

10.88.3.6 ICameraList& operator= ( const ICameraList & ) [protected]

10.88.3.7 virtual CameraPtr operator[] ( unsigned int *index* ) [pure virtual]

Implemented in [CameraList](#).

10.88.3.8 virtual void RemoveByIndex ( unsigned int *index* ) [pure virtual]

Implemented in [CameraList](#).

10.88.3.9 virtual void RemoveBySerial ( std::string *serialNumber* ) [pure virtual]

Implemented in [CameraList](#).

## 10.88.4 Friends And Related Function Documentation

10.88.4.1 friend class CameraListImpl [friend]

10.88.4.2 friend class InterfacelImpl [friend]

## 10.88.5 Member Data Documentation

10.88.5.1 CameraListData\* m\_pCameraListData [protected]

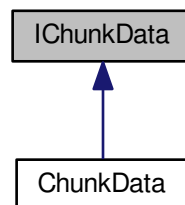
The documentation for this class was generated from the following file:

- include/Interface/ICameraList.h

## 10.89 IChunkData Class Reference

The [Interface](#) file for [ChunkData](#).

Inheritance diagram for IChunkData:



### Public Member Functions

- virtual [~IChunkData](#) ()
- virtual void [SetChunks](#) ([GenApi::INodeMap](#) &pNodeMap)=0
- virtual [float64\\_t](#) [GetBlackLevel](#) () const =0
- virtual [int64\\_t](#) [GetFrameID](#) () const =0
- virtual [float64\\_t](#) [GetExposureTime](#) () const =0
- virtual [int64\\_t](#) [GetTimestamp](#) () const =0
- virtual [int64\\_t](#) [GetExposureEndLineStatusAll](#) () const =0
- virtual [int64\\_t](#) [GetWidth](#) () const =0
- virtual [int64\\_t](#) [GetImage](#) () const =0
- virtual [int64\\_t](#) [GetHeight](#) () const =0
- virtual [float64\\_t](#) [GetGain](#) () const =0
- virtual [int64\\_t](#) [GetSequencerSetActive](#) () const =0
- virtual [int64\\_t](#) [GetCRC](#) () const =0
- virtual [int64\\_t](#) [GetOffsetX](#) () const =0
- virtual [int64\\_t](#) [GetOffsetY](#) () const =0
- virtual [int64\\_t](#) [GetSerialDataLength](#) () const =0
- virtual [int64\\_t](#) [GetPartSelector](#) () const =0
- virtual [int64\\_t](#) [GetPixelDynamicRangeMin](#) () const =0
- virtual [int64\\_t](#) [GetPixelDynamicRangeMax](#) () const =0
- virtual [int64\\_t](#) [GetTimestampLatchValue](#) () const =0
- virtual [int64\\_t](#) [GetLineStatusAll](#) () const =0
- virtual [int64\\_t](#) [GetCounterValue](#) () const =0
- virtual [float64\\_t](#) [GetTimerValue](#) () const =0
- virtual [int64\\_t](#) [GetScanLineSelector](#) () const =0
- virtual [int64\\_t](#) [GetEncoderValue](#) () const =0
- virtual [int64\\_t](#) [GetLinePitch](#) () const =0
- virtual [int64\\_t](#) [GetTransferBlockID](#) () const =0
- virtual [int64\\_t](#) [GetTransferQueueCurrentBlockCount](#) () const =0
- virtual [int64\\_t](#) [GetStreamChannelID](#) () const =0
- virtual [float64\\_t](#) [GetScan3dCoordinateScale](#) () const =0
- virtual [float64\\_t](#) [GetScan3dCoordinateOffset](#) () const =0

- virtual [float64\\_t GetScan3dInvalidDataValue](#) () const =0
- virtual [float64\\_t GetScan3dAxisMin](#) () const =0
- virtual [float64\\_t GetScan3dAxisMax](#) () const =0
- virtual [float64\\_t GetScan3dTransformValue](#) () const =0
- virtual [float64\\_t GetScan3dCoordinateReferenceValue](#) () const =0
- virtual [int64\\_t GetInferenceResult](#) () const =0
- virtual [float64\\_t GetInferenceConfidence](#) () const =0

## Protected Member Functions

- [IChunkData](#) ()

## 10.89.1 Detailed Description

The [Interface](#) file for [ChunkData](#).

## 10.89.2 Constructor & Destructor Documentation

10.89.2.1 virtual [~IChunkData](#) ( ) [inline],[virtual]

10.89.2.2 [IChunkData](#) ( ) [inline],[protected]

## 10.89.3 Member Function Documentation

10.89.3.1 virtual [float64\\_t GetBlackLevel](#) ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.2 virtual [int64\\_t GetCounterValue](#) ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.3 virtual [int64\\_t GetCRC](#) ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.4 virtual [int64\\_t GetEncoderValue](#) ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.5 virtual [int64\\_t GetExposureEndLineStatusAll](#) ( ) const [pure virtual]

Implemented in [ChunkData](#).



10.89.3.6 `virtual float64_t GetExposureTime ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.7 `virtual int64_t GetFrameID ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.8 `virtual float64_t GetGain ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.9 `virtual int64_t GetHeight ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.10 `virtual int64_t GetImage ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.11 `virtual float64_t GetInferenceConfidence ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.12 `virtual int64_t GetInferenceResult ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.13 `virtual int64_t GetLinePitch ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.14 `virtual int64_t GetLineStatusAll ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.15 `virtual int64_t GetOffsetX ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.16 `virtual int64_t GetOffsetY ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.17 `virtual int64_t GetPartSelector ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.18 `virtual int64_t GetPixelDynamicRangeMax ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.19 `virtual int64_t GetPixelDynamicRangeMin ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.20 `virtual float64_t GetScan3dAxisMax ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.21 `virtual float64_t GetScan3dAxisMin ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.22 `virtual float64_t GetScan3dCoordinateOffset ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.23 `virtual float64_t GetScan3dCoordinateReferenceValue ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.24 `virtual float64_t GetScan3dCoordinateScale ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.25 `virtual float64_t GetScan3dInvalidDataValue ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.26 virtual float64\_t GetScan3dTransformValue ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.27 virtual int64\_t GetScanLineSelector ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.28 virtual int64\_t GetSequencerSetActive ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.29 virtual int64\_t GetSerialDataLength ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.30 virtual int64\_t GetStreamChannelID ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.31 virtual float64\_t GetTimerValue ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.32 virtual int64\_t GetTimestamp ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.33 virtual int64\_t GetTimestampLatchValue ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.34 virtual int64\_t GetTransferBlockID ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.35 virtual int64\_t GetTransferQueueCurrentBlockCount ( ) const [pure virtual]

Implemented in [ChunkData](#).

10.89.3.36 `virtual int64_t GetWidth ( ) const [pure virtual]`

Implemented in [ChunkData](#).

10.89.3.37 `virtual void SetChunks ( GenApi::INodeMap & pNodeMap ) [pure virtual]`

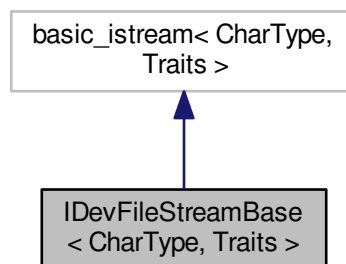
Implemented in [ChunkData](#).

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

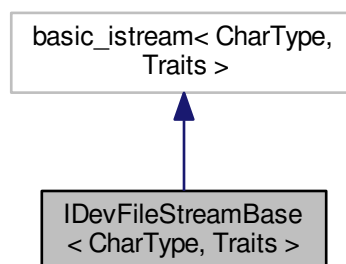
- `include/Interface/IChunkData.h`

## 10.90 IDevFileStreamBase< CharType, Traits > Class Template Reference

Inheritance diagram for IDevFileStreamBase< CharType, Traits >:



Collaboration diagram for IDevFileStreamBase< CharType, Traits >:



## Public Types

- typedef IDevFileStreamBuf< CharType, Traits > [filebuf\\_type](#)
- typedef std::basic\_ios< CharType, Traits > [ios\\_type](#)
- typedef std::basic\_istream< CharType, Traits > [istream\\_type](#)

## Public Member Functions

- [filebuf\\_type](#) \* [rdbuf](#) () const
- bool [is\\_open](#) () const
- void [open](#) (Spinnaker::GenApi::INodeMap \*pInterface, const char \*pFileName, std::ios\_base::openmode mode=std::ios\_base::in)  
*Open file on device in write mode.*
- void [close](#) ()  
*Close the file on the device.*

### 10.90.1 Member Typedef Documentation

10.90.1.1 typedef IDevFileStreamBuf<CharType, Traits> [filebuf\\_type](#)

10.90.1.2 typedef std::basic\_ios<CharType, Traits> [ios\\_type](#)

10.90.1.3 typedef std::basic\_istream<CharType, Traits> [istream\\_type](#)

### 10.90.2 Member Function Documentation

10.90.2.1 void [close](#) ( ) [\[inline\]](#)

Close the file on the device.

10.90.2.2 bool [is\\_open](#) ( ) const [\[inline\]](#)

10.90.2.3 void [open](#) ( Spinnaker::GenApi::INodeMap \* *pInterface*, const char \* *pFileName*, std::ios\_base::openmode *mode* = std::ios\_base::in ) [\[inline\]](#)

Open file on device in write mode.

#### Parameters

|                   |                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------|
| <i>pInterface</i> | <a href="#">NodeMap</a> of the device to which the <a href="#">FileProtocolAdapter</a> is attached |
| <i>pFileName</i>  | Name of the file to open                                                                           |
| <i>mode</i>       | open mode                                                                                          |

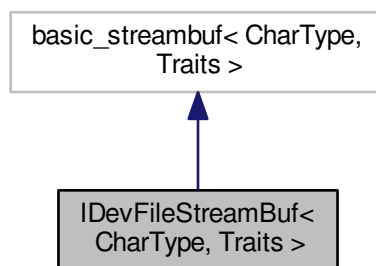
10.90.2.4 `filebuf_type* rdbuf ( ) const` `[inline]`

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

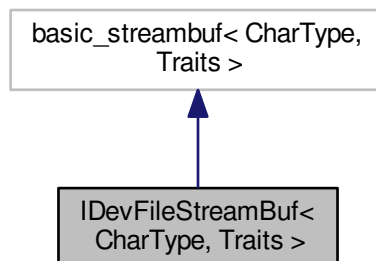
- `include/SpinGenApi/Filestream.h`

## 10.91 IDevFileStreamBuf< CharType, Traits > Class Template Reference

Inheritance diagram for IDevFileStreamBuf< CharType, Traits >:



Collaboration diagram for IDevFileStreamBuf< CharType, Traits >:



### Public Member Functions

- `IDevFileStreamBuf ( )`
- `~IDevFileStreamBuf ( )`
- `filebuf_type * open (Spinnaker::GenApi::INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::in)`
- `bool is_open ( ) const`
- `filebuf_type * close ( )`

### Protected Member Functions

- `int_type underflow ()`
- `int_type pbackfail (int_type c)`

### 10.91.1 Constructor & Destructor Documentation

10.91.1.1 `IDevFileStreamBuf ( )` `[inline]`

10.91.1.2 `~IDevFileStreamBuf ( )` `[inline]`

### 10.91.2 Member Function Documentation

10.91.2.1 `filebuf_type* close ( )` `[inline]`

10.91.2.2 `bool is_open ( ) const` `[inline]`

10.91.2.3 `filebuf_type* open ( Spinnaker::GenApi::INodeMap * pInterface, const char * pFileName, std::ios_base::openmode mode = std::ios_base::in )` `[inline]`

10.91.2.4 `int_type pbackfail ( int_type c )` `[inline]`, `[protected]`

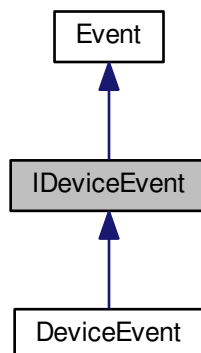
10.91.2.5 `int_type underflow ( )` `[inline]`, `[protected]`

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

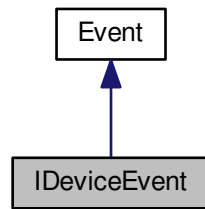
- `include/SpinGenApi/Filestream.h`

## 10.92 IDeviceEvent Class Reference

Inheritance diagram for IDeviceEvent:



Collaboration diagram for IDeviceEvent:



### Public Member Functions

- virtual `~IDeviceEvent()`
- virtual void `OnDeviceEvent(Spinnaker::GenICam::gcstring eventName)=0`
- virtual uint64\_t `GetDeviceEventId()` const =0
- virtual `GenICam::gcstring GetDeviceEventName()` const =0

### Protected Member Functions

- `IDeviceEvent()`
- `IDeviceEvent(const IDeviceEvent &)`
- `IDeviceEvent & operator= (const IDeviceEvent &)`

### Additional Inherited Members

#### 10.92.1 Constructor & Destructor Documentation

10.92.1.1 virtual `~IDeviceEvent()` [inline],[virtual]

10.92.1.2 `IDeviceEvent()` [inline],[protected]

10.92.1.3 `IDeviceEvent(const IDeviceEvent &)` [inline],[protected]

#### 10.92.2 Member Function Documentation

10.92.2.1 virtual uint64\_t `GetDeviceEventId()` const [pure virtual]

Implemented in [DeviceEvent](#).

10.92.2.2 virtual `GenICam::gcstring GetDeviceEventName()` const [pure virtual]

Implemented in [DeviceEvent](#).



10.92.2.3 virtual void OnDeviceEvent ( Spinnaker::GenICam::gcstring eventName ) [pure virtual]

Implemented in [DeviceEvent](#).

10.92.2.4 IDeviceEvent& operator= ( const IDeviceEvent & ) [protected]

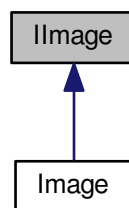
The documentation for this class was generated from the following file:

- include/Interface/[IDeviceEvent.h](#)

## 10.93 IImage Class Reference

The interface file for [Image](#).

Inheritance diagram for IImage:



### Public Member Functions

- virtual [~IImage](#) ()
- virtual [ColorProcessingAlgorithm](#) GetColorProcessing () const =0
- virtual [ImagePtr](#) Convert (Spinnaker::PixelFormatEnums format, [ColorProcessingAlgorithm](#) color↔Algorithm=DEFAULT) const =0
- virtual [ImagePtr](#) ExtractPolarization (const [PolarizationAlgorithm](#) polarizationAlgorithm, const [Polarization↔Resolution](#) resolution) const =0
- virtual void [ResetImage](#) (size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [Spinnaker::PixelFormat↔Enums](#) pixelFormat)=0
- virtual void [ResetImage](#) (size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [Spinnaker::PixelFormat↔Enums](#) pixelFormat, void \*pData)=0
- virtual void [Release](#) ()=0
- virtual uint64\_t [GetID](#) () const =0
- virtual void \* [GetData](#) () const =0
- virtual void \* [GetPrivateData](#) () const =0
- virtual size\_t [GetBufferSize](#) () const =0
- virtual void [DeepCopy](#) (const [ImagePtr](#) pSrcImage)=0
- virtual size\_t [GetWidth](#) () const =0

- virtual size\_t [GetHeight](#) () const =0
- virtual size\_t [GetStride](#) () const =0
- virtual size\_t [GetBitsPerPixel](#) () const =0
- virtual size\_t [GetNumChannels](#) () const =0
- virtual size\_t [GetXOffset](#) () const =0
- virtual size\_t [GetYOffset](#) () const =0
- virtual size\_t [GetXPadding](#) () const =0
- virtual size\_t [GetYPadding](#) () const =0
- virtual uint64\_t [GetFrameID](#) () const =0
- virtual size\_t [GetPayloadType](#) () const =0
- virtual [PayloadTypeInfoDs](#) [GetTLPayloadType](#) () const =0
- virtual uint64\_t [GetTLPixelFormat](#) () const =0
- virtual [PixelFormatNamespaceID](#) [GetTLPixelFormatNamespace](#) () const =0
- virtual [GenICam::gcstring](#) [GetPixelFormatName](#) () const =0
- virtual [Spinnaker::PixelFormatEnums](#) [GetPixelFormat](#) () const =0
- virtual [Spinnaker::PixelFormatIntType](#) [GetPixelFormatIntType](#) () const =0
- virtual bool [IsIncomplete](#) () const =0
- virtual size\_t [GetValidPayloadSize](#) () const =0
- virtual uint64\_t [GetChunkLayoutId](#) () const =0
- virtual uint64\_t [GetTimeStamp](#) () const =0
- virtual void [Save](#) (const char \*pFilename, [ImageFileFormat](#) format=[FROM\\_FILE\\_EXT](#))=0
- virtual void [Save](#) (const char \*pFilename, [PNGOption](#) &pOption)=0
- virtual void [Save](#) (const char \*pFilename, [PPMOption](#) &pOption)=0
- virtual void [Save](#) (const char \*pFilename, [PGMOption](#) &pOption)=0
- virtual void [Save](#) (const char \*pFilename, [TIFFOption](#) &pOption)=0
- virtual void [Save](#) (const char \*pFilename, [JPEGOption](#) &pOption)=0
- virtual void [Save](#) (const char \*pFilename, [JPG2Option](#) &pOption)=0
- virtual void [Save](#) (const char \*pFilename, [BMPOption](#) &pOption)=0
- virtual const [ChunkData](#) & [GetChunkData](#) () const =0
- virtual void [CalculateStatistics](#) ([ImageStatistics](#) &pStatistics)=0
- virtual bool [HasCRC](#) () const =0
- virtual bool [CheckCRC](#) () const =0
- virtual size\_t [GetImageSize](#) () const =0
- virtual bool [IsInUse](#) ()=0
- virtual [ImageStatus](#) [GetImageStatus](#) () const =0
- virtual float \* [GetPolarizationValues](#) () const =0
- virtual [PolarizationAlgorithm](#) [GetPolarizationAlgorithm](#) () const =0

## Protected Member Functions

- [IImage](#) ()

### 10.93.1 Detailed Description

The interface file for [Image](#).

## 10.93.2 Constructor & Destructor Documentation

10.93.2.1 `virtual ~Image ( ) [inline],[virtual]`

10.93.2.2 `Image ( ) [inline],[protected]`

## 10.93.3 Member Function Documentation

10.93.3.1 `virtual void CalculateStatistics ( ImageStatistics & pStatistics ) [pure virtual]`

Implemented in [Image](#).

10.93.3.2 `virtual bool CheckCRC ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.3 `virtual ImagePtr Convert ( Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.4 `virtual void DeepCopy ( const ImagePtr pSrcImage ) [pure virtual]`

Implemented in [Image](#).

10.93.3.5 `virtual ImagePtr ExtractPolarization ( const PolarizationAlgorithm polarizationAlogrithm, const PolarizationResolution resolution ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.6 `virtual size_t GetBitsPerPixel ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.7 `virtual size_t GetBufferSize ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.8 `virtual const ChunkData& GetChunkData ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.9 `virtual uint64_t GetChunkLayoutId ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.10 `virtual ColorProcessingAlgorithm GetColorProcessing ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.11 `virtual void* GetData ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.12 `virtual uint64_t GetFrameID ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.13 `virtual size_t GetHeight ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.14 `virtual uint64_t GetID ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.15 `virtual size_t GetImageSize ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.16 `virtual ImageStatus GetImageStatus ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.17 `virtual size_t GetNumChannels ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.18 `virtual size_t GetPayloadType ( ) const [pure virtual]`

Implemented in [Image](#).

10.93.3.19 virtual **Spinnaker::PixelFormatEnums** GetPixelFormat ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.20 virtual **Spinnaker::PixelFormatIntType** GetPixelFormatIntType ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.21 virtual **GenICam::gcstring** GetPixelFormatName ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.22 virtual **PolarizationAlgorithm** GetPolarizationAlgorithm ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.23 virtual float\* GetPolarizationValues ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.24 virtual void\* GetPrivateData ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.25 virtual size\_t GetStride ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.26 virtual uint64\_t GetTimeStamp ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.27 virtual **PayloadTypeInfoIDs** GetTLPayloadType ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.28 virtual uint64\_t GetTLPixelFormat ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.29 virtual PixelFormatNamespaceID GetTLPixelFormatNamespace ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.30 virtual size\_t GetValidPayloadSize ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.31 virtual size\_t GetWidth ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.32 virtual size\_t GetXOffset ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.33 virtual size\_t GetXPadding ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.34 virtual size\_t GetYOffset ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.35 virtual size\_t GetYPadding ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.36 virtual bool HasCRC ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.37 virtual bool IsIncomplete ( ) const [pure virtual]

Implemented in [Image](#).

10.93.3.38 virtual bool IsInUse ( ) [pure virtual]

Implemented in [Image](#).

10.93.3.39 virtual void Release ( ) [pure virtual]

Implemented in [Image](#).

10.93.3.40 virtual void ResetImage ( size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, Spinnaker::PixelFormatEnums pixelFormat ) [pure virtual]

Implemented in [Image](#).

10.93.3.41 virtual void ResetImage ( size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void \* pData ) [pure virtual]

Implemented in [Image](#).

10.93.3.42 virtual void Save ( const char \* pFilename, ImageFileFormat format = FROM\_FILE\_EXT ) [pure virtual]

Implemented in [Image](#).

10.93.3.43 virtual void Save ( const char \* pFilename, PNGOption & pOption ) [pure virtual]

Implemented in [Image](#).

10.93.3.44 virtual void Save ( const char \* pFilename, PPMOption & pOption ) [pure virtual]

Implemented in [Image](#).

10.93.3.45 virtual void Save ( const char \* pFilename, PGMOption & pOption ) [pure virtual]

Implemented in [Image](#).

10.93.3.46 virtual void Save ( const char \* pFilename, TIFFOption & pOption ) [pure virtual]

Implemented in [Image](#).

10.93.3.47 virtual void Save ( const char \* pFilename, JPEGOption & pOption ) [pure virtual]

Implemented in [Image](#).

10.93.3.48 virtual void Save ( const char \* pFilename, JPG2Option & pOption ) [pure virtual]

Implemented in [Image](#).

10.93.3.49 `virtual void Save ( const char * pFilename, BMPOption & pOption )` [pure virtual]

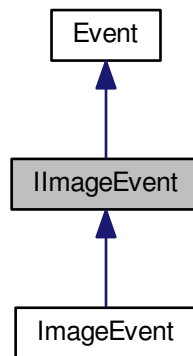
Implemented in [Image](#).

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

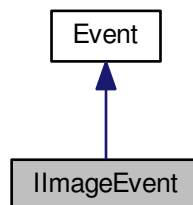
- [include/Interface/IImage.h](#)

## 10.94 IImageEvent Class Reference

Inheritance diagram for IImageEvent:



Collaboration diagram for IImageEvent:



### Public Member Functions

- virtual `~IImageEvent ()`
- virtual void `OnImageEvent (ImagePtr image)=0`



### Protected Member Functions

- [`ImageEvent`](#) ()
- [`ImageEvent`](#) (const [`ImageEvent`](#) &)
- [`ImageEvent`](#) & [`operator=`](#) (const [`ImageEvent`](#) &)

### Additional Inherited Members

#### 10.94.1 Constructor & Destructor Documentation

10.94.1.1 `virtual ~ImageEvent ( ) [inline],[virtual]`

10.94.1.2 `ImageEvent ( ) [inline],[protected]`

10.94.1.3 `ImageEvent ( const ImageEvent & ) [inline],[protected]`

#### 10.94.2 Member Function Documentation

10.94.2.1 `virtual void OnImageEvent ( ImagePtr image ) [pure virtual]`

Implemented in [`ImageEvent`](#).

10.94.2.2 `ImageEvent& operator= ( const ImageEvent & ) [protected]`

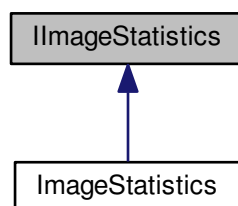
The documentation for this class was generated from the following file:

- `include/Interface/ImageEvent.h`

## 10.95 `ImageStatistics` Class Reference

The interface file for image statistics.

Inheritance diagram for `ImageStatistics`:



## Public Member Functions

- virtual [~IImageStatistics](#) ()
- virtual void [EnableAll](#) ()=0
- virtual void [DisableAll](#) ()=0
- virtual void [EnableGreyOnly](#) ()=0
- virtual void [EnableRGBOnly](#) ()=0
- virtual void [EnableHSLOnly](#) ()=0
- virtual void [GetChannelStatus](#) ([StatisticsChannel](#) channel, bool \*pEnabled) const =0
- virtual void [SetChannelStatus](#) ([StatisticsChannel](#) channel, bool enabled)=0
- virtual void [GetRange](#) ([StatisticsChannel](#) channel, unsigned int \*pMin, unsigned int \*pMax) const =0
- virtual void [GetPixelValueRange](#) ([StatisticsChannel](#) channel, unsigned int \*pPixelValueMin, unsigned int \*pPixelValueMax) const =0
- virtual void [GetNumPixelValues](#) ([StatisticsChannel](#) channel, unsigned int \*pNumPixelValues) const =0
- virtual void [GetMean](#) ([StatisticsChannel](#) channel, float \*pPixelValueMean) const =0
- virtual void [GetHistogram](#) ([StatisticsChannel](#) channel, int \*\*ppHistogram) const =0
- virtual void [GetStatistics](#) ([StatisticsChannel](#) channel, unsigned int \*pRangeMin=NULL, unsigned int \*pRangeMax=NULL, unsigned int \*pPixelValueMin=NULL, unsigned int \*pPixelValueMax=NULL, unsigned int \*pNumPixelValues=NULL, float \*pPixelValueMean=NULL, int \*\*ppHistogram=NULL) const =0

## Protected Member Functions

- [IImageStatistics](#) ()
- [IImageStatistics](#) (const [IImageStatistics](#) &)

### 10.95.1 Detailed Description

The interface file for image statistics.

### 10.95.2 Constructor & Destructor Documentation

10.95.2.1 virtual [~IImageStatistics](#) ( ) `[inline], [virtual]`

10.95.2.2 [IImageStatistics](#) ( ) `[inline], [protected]`

10.95.2.3 [IImageStatistics](#) ( const [IImageStatistics](#) & ) `[inline], [protected]`

### 10.95.3 Member Function Documentation

10.95.3.1 virtual void [DisableAll](#) ( ) `[pure virtual]`

Implemented in [ImageStatistics](#).

10.95.3.2 virtual void [EnableAll](#) ( ) `[pure virtual]`

Implemented in [ImageStatistics](#).

10.95.3.3 virtual void EnableGreyOnly ( ) [pure virtual]

Implemented in [ImageStatistics](#).

10.95.3.4 virtual void EnableHSLOnly ( ) [pure virtual]

Implemented in [ImageStatistics](#).

10.95.3.5 virtual void EnableRGBOnly ( ) [pure virtual]

Implemented in [ImageStatistics](#).

10.95.3.6 virtual void GetChannelStatus ( **StatisticsChannel** *channel*, bool \* *pEnabled* ) const [pure virtual]

Implemented in [ImageStatistics](#).

10.95.3.7 virtual void GetHistogram ( **StatisticsChannel** *channel*, int \*\* *ppHistogram* ) const [pure virtual]

Implemented in [ImageStatistics](#).

10.95.3.8 virtual void GetMean ( **StatisticsChannel** *channel*, float \* *pPixelValueMean* ) const [pure virtual]

Implemented in [ImageStatistics](#).

10.95.3.9 virtual void GetNumPixelValues ( **StatisticsChannel** *channel*, unsigned int \* *pNumPixelValues* ) const [pure virtual]

Implemented in [ImageStatistics](#).

10.95.3.10 virtual void GetPixelValueRange ( **StatisticsChannel** *channel*, unsigned int \* *pPixelValueMin*, unsigned int \* *pPixelValueMax* ) const [pure virtual]

Implemented in [ImageStatistics](#).

10.95.3.11 virtual void GetRange ( **StatisticsChannel** *channel*, unsigned int \* *pMin*, unsigned int \* *pMax* ) const [pure virtual]

Implemented in [ImageStatistics](#).

```

10.95.3.12 virtual void GetStatistics ( StatisticsChannel channel, unsigned int * pRangeMin = NULL, unsigned int *
        pRangeMax = NULL, unsigned int * pPixelValueMin = NULL, unsigned int * pPixelValueMax = NULL, unsigned
        int * pNumPixelValues = NULL, float * pPixelValueMean = NULL, int ** ppHistogram = NULL ) const [pure
        virtual]

```

Implemented in [ImageStatistics](#).

```

10.95.3.13 virtual void SetChannelStatus ( StatisticsChannel channel, bool enabled ) [pure virtual]

```

Implemented in [ImageStatistics](#).

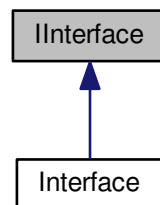
The documentation for this class was generated from the following file:

- [include/Interface/IImageStatistics.h](#)

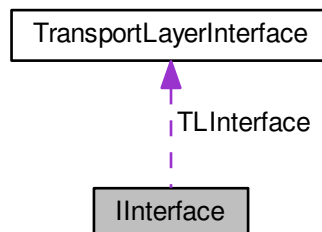
## 10.96 IInterface Class Reference

The interface file for [Interface](#).

Inheritance diagram for IInterface:



Collaboration diagram for IInterface:



## Public Member Functions

- virtual [~IInterface](#) ()
- virtual [CameraList GetCameras](#) (bool updateCameras=true) const =0
- virtual bool [UpdateCameras](#) ()=0
- virtual [GenApi::INodeMap & GetTLNodeMap](#) () const =0
- virtual void [RegisterEvent](#) ([Event](#) &evtToRegister)=0
- virtual void [UnregisterEvent](#) ([Event](#) &evtToUnregister)=0
- virtual bool [IsInUse](#) () const =0
- virtual void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int \*pResultSize=0, [ActionCommandResult](#) results[]=NULL) const =0

## Public Attributes

- [TransportLayerInterface TLInterface](#)

## Protected Member Functions

- [IInterface](#) ()
- [IInterface](#) (const [IInterface](#) &)
- [IInterface](#) & [operator=](#) (const [IInterface](#) &)

## Protected Attributes

- InterfaceData \* [m\\_pInterfaceData](#)

## Friends

- class [InterfaceInternal](#)

### 10.96.1 Detailed Description

The interface file for [Interface](#).

### 10.96.2 Constructor & Destructor Documentation

10.96.2.1 virtual [~IInterface](#) ( ) [inline],[virtual]

10.96.2.2 [IInterface](#) ( ) [inline],[protected]

10.96.2.3 [IInterface](#) ( const [IInterface](#) & ) [inline],[protected]

### 10.96.3 Member Function Documentation

10.96.3.1 virtual [CameraList GetCameras](#) ( bool *updateCameras* =true ) const [pure virtual]

Implemented in [Interface](#).

10.96.3.2 virtual **GenApi::INodeMap& GetTLNodeMap** ( ) const [pure virtual]

Implemented in [Interface](#).

10.96.3.3 virtual bool **IsInUse** ( ) const [pure virtual]

Implemented in [Interface](#).

10.96.3.4 **Interface& operator=** ( const **Interface** & ) [protected]

10.96.3.5 virtual void **RegisterEvent** ( **Event** & *evtToRegister* ) [pure virtual]

Implemented in [Interface](#).

10.96.3.6 virtual void **SendActionCommand** ( unsigned int *deviceKey*, unsigned int *groupKey*, unsigned int *groupMask*, unsigned long long *actionTime* = 0, unsigned int \* *pResultSize* = 0, **ActionCommandResult** *results*[] = NULL ) const [pure virtual]

Implemented in [Interface](#).

10.96.3.7 virtual void **UnregisterEvent** ( **Event** & *evtToUnregister* ) [pure virtual]

Implemented in [Interface](#).

10.96.3.8 virtual bool **UpdateCameras** ( ) [pure virtual]

Implemented in [Interface](#).

## 10.96.4 Friends And Related Function Documentation

10.96.4.1 friend class **InterfaceInternal** [friend]

## 10.96.5 Member Data Documentation

10.96.5.1 **InterfaceData\*** *m\_pInterfaceData* [protected]

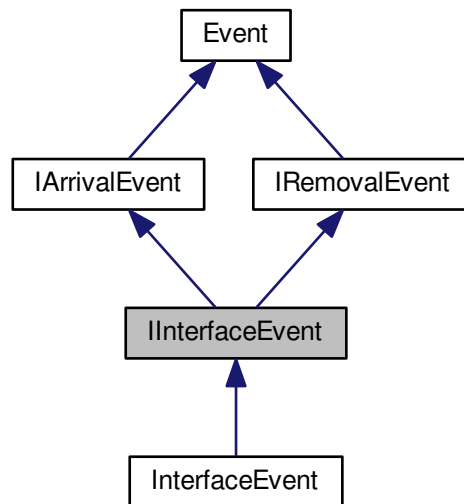
10.96.5.2 **TransportLayerInterface** **TLInterface**

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

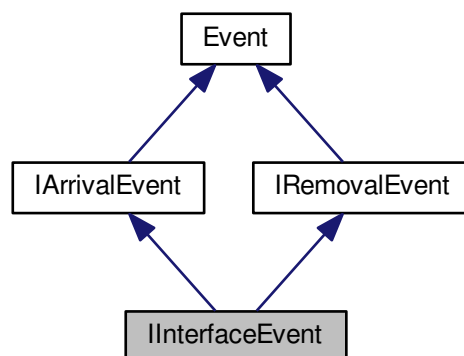
- include/Interface/[Interface.h](#)

## 10.97 IInterfaceEvent Class Reference

Inheritance diagram for IInterfaceEvent:



Collaboration diagram for IInterfaceEvent:



### Public Member Functions

- virtual [~IInterfaceEvent](#) ()
- virtual void [OnDeviceArrival](#) (uint64\_t serialNumber)=0
- virtual void [OnDeviceRemoval](#) (uint64\_t serialNumber)=0

## Protected Member Functions

- [InterfaceEvent](#) ()
- [InterfaceEvent](#) (const [InterfaceEvent](#) &)
- [InterfaceEvent](#) & [operator=](#) (const [InterfaceEvent](#) &)

## Additional Inherited Members

### 10.97.1 Constructor & Destructor Documentation

10.97.1.1 `virtual ~InterfaceEvent ( ) [inline],[virtual]`

10.97.1.2 `InterfaceEvent ( ) [inline],[protected]`

10.97.1.3 `InterfaceEvent ( const InterfaceEvent & ) [inline],[protected]`

### 10.97.2 Member Function Documentation

10.97.2.1 `virtual void OnDeviceArrival ( uint64_t serialNumber ) [pure virtual]`

Implements [IArrivalEvent](#).

Implemented in [InterfaceEvent](#).

10.97.2.2 `virtual void OnDeviceRemoval ( uint64_t serialNumber ) [pure virtual]`

Implements [IRemovalEvent](#).

Implemented in [InterfaceEvent](#).

10.97.2.3 `InterfaceEvent& operator= ( const InterfaceEvent & ) [protected]`

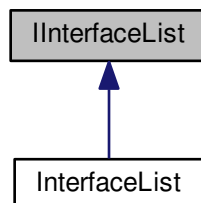
The documentation for this class was generated from the following file:

- `include/Interface/InterfaceEvent.h`

## 10.98 InterfaceList Class Reference

The interface file for [InterfaceList](#) class.

Inheritance diagram for [InterfaceList](#):





## Public Member Functions

- virtual [~IList](#) (void)
- virtual [IntPtr operator\[\]](#) (unsigned int index)=0
- virtual unsigned int [GetSize](#) () const =0
- virtual [IntPtr GetByIndex](#) (unsigned int index) const =0
- virtual void [Clear](#) ()=0

## Protected Member Functions

- [IList](#) (void)
- [IList](#) (const [IList](#) &)
- [IList](#) & [operator=](#) (const [IList](#) &)

## Protected Attributes

- [IListData](#) \* [m\\_pIListData](#)

### 10.98.1 Detailed Description

The interface file for [IList](#) class.

### 10.98.2 Constructor & Destructor Documentation

10.98.2.1 virtual [~IList](#) ( void ) [inline],[virtual]

10.98.2.2 [IList](#) ( void ) [inline],[protected]

10.98.2.3 [IList](#) ( const [IList](#) & ) [inline],[protected]

### 10.98.3 Member Function Documentation

10.98.3.1 virtual void [Clear](#) ( ) [pure virtual]

Implemented in [IList](#).

10.98.3.2 virtual [IntPtr](#) [GetByIndex](#) ( unsigned int *index* ) const [pure virtual]

Implemented in [IList](#).

10.98.3.3 virtual unsigned int [GetSize](#) ( ) const [pure virtual]

Implemented in [IList](#).

10.98.3.4 `InterfaceList& operator= ( const InterfaceList & )` [protected]

10.98.3.5 `virtual InterfacePtr operator[] ( unsigned int index )` [pure virtual]

Implemented in [InterfaceList](#).

## 10.98.4 Member Data Documentation

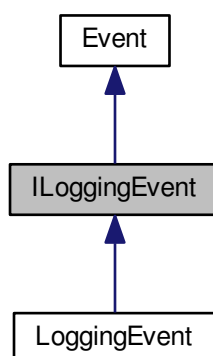
10.98.4.1 `InterfaceListData* m_pInterfaceListData` [protected]

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

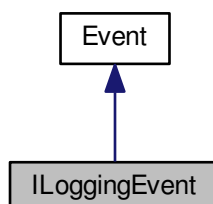
- `include/Interface/InterfaceList.h`

## 10.99 ILoggingEvent Class Reference

Inheritance diagram for ILoggingEvent:



Collaboration diagram for ILoggingEvent:



## Public Member Functions

- virtual [~ILoggingEvent](#) ()
- virtual void [OnLogEvent](#) ([LoggingEventDataPtr](#) eventPtr)=0

## Protected Member Functions

- [ILoggingEvent](#) ()
- [ILoggingEvent](#) (const [ILoggingEvent](#) &)
- [ILoggingEvent](#) & [operator=](#) (const [ILoggingEvent](#) &)

## Additional Inherited Members

### 10.99.1 Constructor & Destructor Documentation

10.99.1.1 virtual [~ILoggingEvent](#) ( ) [inline],[virtual]

10.99.1.2 [ILoggingEvent](#) ( ) [inline],[protected]

10.99.1.3 [ILoggingEvent](#) ( const [ILoggingEvent](#) & ) [inline],[protected]

### 10.99.2 Member Function Documentation

10.99.2.1 virtual void [OnLogEvent](#) ( [LoggingEventDataPtr](#) *eventPtr* ) [pure virtual]

Implemented in [LoggingEvent](#).

10.99.2.2 [ILoggingEvent&](#) [operator=](#) ( const [ILoggingEvent](#) & ) [protected]

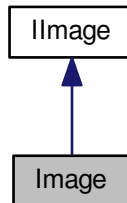
The documentation for this class was generated from the following file:

- include/Interface/[ILoggingEvent.h](#)

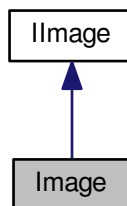
## 10.100 Image Class Reference

The image object class.

Inheritance diagram for Image:



Collaboration diagram for Image:



### Public Member Functions

- virtual [~Image](#) ()  
*Virtual destructor.*
- [ColorProcessingAlgorithm GetColorProcessing](#) () const  
*Gets the color algorithm used to produce the image.*
- [ImagePtr Convert](#) ([Spinnaker::PixelFormatEnums](#) format, [ColorProcessingAlgorithm](#) colorAlgorithm=[DEFAULT](#)) const  
*Converts the current image buffer to the specified output pixel format and stores the result in the specified image.*
- void [ResetImage](#) (size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [Spinnaker::PixelFormatEnums](#) pixelFormat)  
*Sets new dimensions of the image object and allocates memory.*
- void [ResetImage](#) (size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [Spinnaker::PixelFormatEnums](#) pixelFormat, void \*pData)  
*Sets new dimensions of the image object.*
- void [Release](#) ()

- `uint64_t GetID () const`  
*Gets a unique ID for this image.*
- `void * GetData () const`  
*Gets a pointer to the data associated with the image.*
- `void * GetPrivateData () const`  
*Gets a pointer to the user passed data associated with the image.*
- `size_t GetBufferSize () const`  
*Gets the size of the buffer associated with the image in bytes.*
- `void DeepCopy (const ImagePtr pSrcImage)`  
*Performs a deep copy of the *Image*.*
- `size_t GetWidth () const`  
*Gets the width of the image in pixels.*
- `size_t GetHeight () const`  
*Gets the height of the image in pixels.*
- `size_t GetStride () const`  
*Gets the stride of the image in bytes.*
- `size_t GetBitsPerPixel () const`  
*Gets the number of bits used per pixel in the image.*
- `size_t GetNumChannels () const`  
*Gets the number of channels (depth) used in the image.*
- `size_t GetXOffset () const`  
*Gets the ROI x offset in pixels for this image.*
- `size_t GetYOffset () const`  
*Gets the ROI y offset in pixels for this image.*
- `size_t GetXPadding () const`  
*Gets the x padding in bytes for this image.*
- `size_t GetYPadding () const`  
*Gets the y padding in bytes for this image.*
- `uint64_t GetFrameID () const`  
*Gets the frame ID for this image.*
- `size_t GetPayloadType () const`  
*Gets the payload type that was transmitted.*
- `PayloadTypeInfoIds GetTLPayloadType () const`  
*Gets the GenTL specific payload type that was transmitted.*
- `uint64_t GetTLPixelFormat () const`  
*Gets the pixel format of the image.*
- `PixelFormatNamespaceID GetTLPixelFormatNamespace () const`  
*Returns an enum value that represents the namespace in which this image's TL specific pixel format resides.*
- `GenICam::gcstring GetPixelFormatName () const`  
*Returns a string value that represents this image's pixel format.*
- `Spinnaker::PixelFormatEnums GetPixelFormat () const`  
*Returns an enum value that represents the pixel format of this image.*
- `Spinnaker::PixelFormatIntType GetPixelFormatIntType () const`  
*Returns an enum value that represents the integer type used in the pixel format of this image.*
- `bool IsIncomplete () const`  
*Returns a boolean value indicating if this image was incomplete.*
- `size_t GetValidPayloadSize () const`  
*Returns the size of valid data in the image payload.*
- `uint64_t GetChunkLayoutId () const`  
*Returns the id of the chunk data layout.*
- `uint64_t GetTimeStamp () const`

- Gets the time stamp for the image in nanoseconds.*
- void [Save](#) (const char \*pFilename, [ImageFileFormat](#) format=[FROM\\_FILE\\_EXT](#))  
*Saves the image to the specified file name with the file format specified.*
- void [Save](#) (const char \*pFilename, [PNGOption](#) &pOption)  
*Saves the image to the specified file name with the options specified.*
- void [Save](#) (const char \*pFilename, [PPMOption](#) &pOption)  
*Saves the image to the specified file name with the options specified.*
- void [Save](#) (const char \*pFilename, [PGMOption](#) &pOption)  
*Saves the image to the specified file name with the options specified.*
- void [Save](#) (const char \*pFilename, [TIFFOption](#) &pOption)  
*Saves the image to the specified file name with the options specified.*
- void [Save](#) (const char \*pFilename, [JPEGOption](#) &pOption)  
*Saves the image to the specified file name with the options specified.*
- void [Save](#) (const char \*pFilename, [JPG2Option](#) &pOption)  
*Saves the image to the specified file name with the options specified.*
- void [Save](#) (const char \*pFilename, [BMPOption](#) &pOption)  
*Saves the image to the specified file name with the options specified.*
- const [ChunkData](#) & [GetChunkData](#) () const  
*Returns a pointer to a chunk data interface.*
- void [CalculateStatistics](#) ([ImageStatistics](#) &pStatistics)  
*Retrieves a number of pixel statistics for an image including a histogram array of the range of pixel values.*
- bool [HasCRC](#) () const  
*Checks if the image contains ImageCRC checksum from chunk data.*
- bool [CheckCRC](#) () const  
*Checks if the computed checksum matches with chunk data's ImageCRC.*
- size\_t [GetImageSize](#) () const  
*Returns the size of the image.*
- bool [IsInUse](#) ()  
*Returns true if the image is still in use by the stream.*
- [ImageStatus](#) [GetImageStatus](#) () const  
*Returns data integrity status of the image returned from [GetNextImage\(\)](#)*
- [ImagePtr](#) [ExtractPolarization](#) (const [PolarizationAlgorithm](#) polarizationAlgorithm, const [PolarizationResolution](#) resolution) const  
*Extracts an image from a monochrome-polarized sensor.*
- float \* [GetPolarizationValues](#) () const  
*Returns the polarization values associated with an extracted polarization image.*
- [PolarizationAlgorithm](#) [GetPolarizationAlgorithm](#) () const  
*Returns the polarization algorithm used to extract a polarization image.*

## Static Public Member Functions

- static [ImagePtr](#) [Create](#) ()  
*Create an image object.*
- static [ImagePtr](#) [Create](#) (const [ImagePtr](#) image)  
*Create an image object that is a deep copy of the input image.*
- static [ImagePtr](#) [Create](#) (size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [Spinnaker::PixelFormat](#)↔  
[Enums](#) pixelFormat, void \*pData)  
*Create an image object with the specified parameters.*
- static void [SetDefaultColorProcessing](#) ([ColorProcessingAlgorithm](#) colorAlgorithm)  
*Sets the default color processing algorithm.*

- static [ColorProcessingAlgorithm](#) [GetDefaultColorProcessing](#) ()  
*Gets the default color processing algorithm.*
- static const char \* [GetImageStatusDescription](#) ([ImageStatus](#) status)  
*Returns a string describing the meaning of the status enum.*
- static void [SetHeatMapColorGradient](#) (const [HeatMapColor](#) newLowColor, const [HeatMapColor](#) newHighColor)  
*Sets the heatmap gradient color vector to the new desired range between HEATMAP\_BLACK and HEATMAP\_WHITE.*
- static void [GetHeatMapColorGradient](#) ([HeatMapColor](#) &currentLowColor, [HeatMapColor](#) &currentHighColor)  
*Returns the current heatmap gradient color range.*
- static void [SetHeatMapRange](#) (const unsigned int newLowValue, const unsigned int newHighValue)  
*Sets the high and low values used to determine which grayscale values are converted to a color 'heatmap' representation.*
- static void [GetHeatMapRange](#) (unsigned int &currentLowValue, unsigned int &currentHighValue)  
*Returns the current high and low values used in heatmap representations.*

### Protected Member Functions

- [Image](#) ()
- [Image](#) (const [ImagePtr](#) image)
- [Image](#) (size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [Spinnaker::PixelFormatEnums](#) pixelFormat, void \*pData)
- [ImagePtr](#) [CreateShared](#) () const
- void [DeepCopy](#) (const [Image](#) &pSrcImage)
- void [Convert](#) ([Spinnaker::PixelFormatEnums](#) format, [Image](#) &pDestImage, [ColorProcessingAlgorithm](#) colorAlgorithm=[DEFAULT](#)) const

### Protected Attributes

- [ImageData](#) \* [m\\_pImageData](#)

### Friends

- class [IDataStream](#)
- class [Stream](#)
- class [ImageConverter](#)
- class [ImageFiler](#)
- class [ImageStatsCalculator](#)

## 10.100.1 Detailed Description

The image object class.

## 10.100.2 Constructor & Destructor Documentation

### 10.100.2.1 virtual ~Image ( ) [virtual]

Virtual destructor.

10.100.2.2 **Image** ( ) [protected]

10.100.2.3 **Image** ( const ImagePtr *image* ) [protected]

10.100.2.4 **Image** ( size\_t *width*, size\_t *height*, size\_t *offsetX*, size\_t *offsetY*, Spinnaker::PixelFormatEnums *pixelFormat*, void \* *pData* ) [protected]

### 10.100.3 Member Function Documentation

10.100.3.1 void **CalculateStatistics** ( ImageStatistics & *pStatistics* ) [virtual]

Retrieves a number of pixel statistics for an image including a histogram array of the range of pixel values.

#### Parameters

|                    |                             |
|--------------------|-----------------------------|
| <i>pStatistics</i> | The statistics of an image. |
|--------------------|-----------------------------|

Implements [Image](#).

10.100.3.2 bool **CheckCRC** ( ) const [virtual]

Checks if the computed checksum matches with chunk data's ImageCRC.

#### Returns

Returns true if computed checksum matches with the chunk data's CRC and false otherwise.

Implements [Image](#).

10.100.3.3 ImagePtr **Convert** ( Spinnaker::PixelFormatEnums *format*, ColorProcessingAlgorithm *colorAlgorithm* = DEFAULT ) const [virtual]

Converts the current image buffer to the specified output pixel format and stores the result in the specified image.

The destination image does not need to be configured in any way before the call is made.

#### See also

[PixelFormatEnums](#)

#### Parameters

|                       |                                                                       |
|-----------------------|-----------------------------------------------------------------------|
| <i>format</i>         | Output format of the converted image.                                 |
| <i>colorAlgorithm</i> | Optional color processing algorithm for producing the converted image |



**Returns**

The converted image.

Implements [Image](#).

**10.100.3.4** `void Convert ( Spinnaker::PixelFormatEnums format, Image & pDestImage, ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const` `[protected]`

**10.100.3.5** `static ImagePtr Create ( )` `[static]`

Create an image object.

**10.100.3.6** `static ImagePtr Create ( const ImagePtr image )` `[static]`

Create an image object that is a deep copy of the input image.

**Parameters**

|              |                         |
|--------------|-------------------------|
| <i>image</i> | The input image to copy |
|--------------|-------------------------|

**10.100.3.7** `static ImagePtr Create ( size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void * pData )` `[static]`

Create an image object with the specified parameters.

**Parameters**

|                    |                            |
|--------------------|----------------------------|
| <i>width</i>       | The image width in pixels  |
| <i>height</i>      | The image height in pixels |
| <i>offsetX</i>     | The image X offset         |
| <i>offsetY</i>     | The image Y offset         |
| <i>pixelFormat</i> | The image pixel format     |
| <i>pData</i>       | The image data             |

**10.100.3.8** `ImagePtr CreateShared ( ) const` `[protected]`

**10.100.3.9** `void DeepCopy ( const ImagePtr pSrcImage )` `[virtual]`

Performs a deep copy of the [Image](#).

After this operation, the image contents and member variables will be the same. The Images will not share a buffer. The [Image](#)'s current buffer will not be released.

## Parameters

|                  |                                                  |
|------------------|--------------------------------------------------|
| <i>pSrcImage</i> | The <a href="#">Image</a> to copy the data from. |
|------------------|--------------------------------------------------|

Implements [Image](#).

10.100.3.10 `void DeepCopy ( const Image & pSrcImage )` `[protected]`

10.100.3.11 `ImagePtr ExtractPolarization ( const PolarizationAlgorithm polarizationAlgorithm, const PolarizationResolution resolution ) const` `[virtual]`

Extracts an image from a monochrome-polarized sensor.

The extracted image will be returned as Mono8 or BGRa8 for heatmap images.

## Parameters

|                              |                                        |
|------------------------------|----------------------------------------|
| <i>polarizationAlgorithm</i> | Desired polarization algorithm to use. |
| <i>resolution</i>            | Desired resolution of output image.    |

## Returns

The converted image.

Implements [Image](#).

10.100.3.12 `size_t GetBitsPerPixel ( ) const` `[virtual]`

Gets the number of bits used per pixel in the image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

## Returns

The number of bits used per pixel.

Implements [Image](#).

10.100.3.13 `size_t GetBufferSize ( ) const` `[virtual]`

Gets the size of the buffer associated with the image in bytes.

## Returns

The size of the buffer, in bytes.

Implements [Image](#).

10.100.3.14 `const ChunkData& GetChunkData ( ) const` [virtual]

Returns a pointer to a chunk data interface.

No ownership is transferred, the chunk data interface reference is valid until [Image::Release\(\)](#) is called on this image.

Returns

[ChunkData](#) interface that provides access to image chunks.

Implements [IImage](#).

10.100.3.15 `uint64_t GetChunkLayoutId ( ) const` [virtual]

Returns the id of the chunk data layout.

Returns

uint64\_t value representing the id of the chunk data layout.

Implements [IImage](#).

10.100.3.16 `ColorProcessingAlgorithm GetColorProcessing ( ) const` [virtual]

Gets the color algorithm used to produce the image.

See also

[Convert\(\)](#)

Returns

The color processing algorithm used to produce the image.

Implements [IImage](#).

10.100.3.17 `void* GetData ( ) const` [virtual]

Gets a pointer to the data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is released. The pointer may also be invalidated if the [Image](#) object is passed to [Image::Release\(\)](#).

Returns

A pointer to the image data.

Implements [IImage](#).

10.100.3.18 `static ColorProcessingAlgorithm GetDefaultColorProcessing ( ) [static]`

Gets the default color processing algorithm.

See also

[SetDefaultColorProcessing\(\)](#)

Returns

The default color processing algorithm.

10.100.3.19 `uint64_t GetFrameID ( ) const [virtual]`

Gets the frame ID for this image.

Returns

The frame ID.

Implements [IImage](#).

10.100.3.20 `static void GetHeatMapColorGradient ( HeatMapColor & currentLowColor, HeatMapColor & currentHighColor ) [static]`

Returns the current heatmap gradient color range.

Heatmap images are currently implemented for polarized cameras only.

Parameters

|                         |                                             |
|-------------------------|---------------------------------------------|
| <i>currentLowColor</i>  | Current color at which the gradient begins. |
| <i>currentHighColor</i> | Current color at which the gradient ends.   |

10.100.3.21 `static void GetHeatMapRange ( unsigned int & currentLowValue, unsigned int & currentHighValue ) [static]`

Returns the current high and low values used in heatmap representations.

Heatmap images are currently implemented for polarized cameras only.

Parameters

|                         |                                                     |
|-------------------------|-----------------------------------------------------|
| <i>currentLowValue</i>  | Current value at which color representation begins. |
| <i>currentHighValue</i> | Current value at which color representation ends.   |

See also

[SetHeatMapRange\(\)](#)

**10.100.3.22** `size_t GetHeight ( ) const [virtual]`

Gets the height of the image in pixels.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

**Returns**

The height in pixels.

Implements [IImage](#).

**10.100.3.23** `uint64_t GetID ( ) const [virtual]`

Gets a unique ID for this image.

Each image in a steam will have a unique ID to help identify it.

**Returns**

The 64 bit unique id for this image.

Implements [IImage](#).

**10.100.3.24** `size_t GetImageSize ( ) const [virtual]`

Returns the size of the image.

**Returns**

The image size in bytes.

Implements [IImage](#).

**10.100.3.25** `ImageStatus GetImageStatus ( ) const [virtual]`

Returns data integrity status of the image returned from `GetNextImage()`

**Returns**

Returns whether image has any data integrity issues.

Implements [IImage](#).

10.100.3.26 `static const char* GetImageStatusDescription ( ImageStatus status )` `[static]`

Returns a string describing the meaning of the status enum.

#### Returns

Returns the meaning of the status enum.

10.100.3.27 `size_t GetNumChannels ( ) const` `[virtual]`

Gets the number of channels (depth) used in the image.

Returns 0 if the number of channels for the given pixel format is unknown.

#### Returns

The number of channels per pixel.

Implements [IImage](#).

10.100.3.28 `size_t GetPayloadType ( ) const` `[virtual]`

Gets the payload type that was transmitted.

This is a device types specific value that identifies how the image was transmitted. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

#### Returns

Device types specific payload type.

Implements [IImage](#).

10.100.3.29 `Spinnaker::PixelFormatEnums GetPixelFormat ( ) const` `[virtual]`

Returns an enum value that represents the pixel format of this image.

The enum can be used with the easy access [GenICam](#) features available through the [Camera.h](#) header file. This easy access enum can also be used in the [Convert\(\)](#) function.

#### See also

[Convert\(\)](#)

#### Returns

enum value representing the PixelFormat.

Implements [IImage](#).

**10.100.3.30 Spinnaker::PixelFormatIntType GetPixelFormatIntType ( ) const** [virtual]

Returns an enum value that represents the integer type used in the pixel format of this image.

**Returns**

enum value representing the integer type used.

Implements [IImage](#).

**10.100.3.31 GenICam::gcstring GetPixelFormatName ( ) const** [virtual]

Returns a string value that represents this image's pixel format.

The string is a valid SFNC name that maps to the underlying TL specific pixel format. This is the most generic way to identify the pixel format of the image.

**Returns**

string value representing the PixelFormat.

Implements [IImage](#).

**10.100.3.32 PolarizationAlgorithm GetPolarizationAlgorithm ( ) const** [virtual]

Returns the polarization algorithm used to extract a polarization image.

**Returns**

The polarization algorithm used to extract the polarization image.

Implements [IImage](#).

**10.100.3.33 float\* GetPolarizationValues ( ) const** [virtual]

Returns the polarization values associated with an extracted polarization image.

Note that standard quadrants (QUADRANT\_I0\_GRAYSCALE - QUADRANT\_I135\_GRAYSCALE) do not provide polarization values.

**Returns**

The polarization values associated with a polarization image.

Implements [IImage](#).

**10.100.3.34** `void* GetPrivateData ( ) const [virtual]`

Gets a pointer to the user passed data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is released. The pointer may also be invalidated if the [Image](#) object is passed to [Image::Release\(\)](#).

TODO: no way to set private data for image yet.

**Returns**

A pointer to the user passed data pointer.

Implements [IImage](#).

**10.100.3.35** `size_t GetStride ( ) const [virtual]`

Gets the stride of the image in bytes.

The stride of an image is how many bytes are in each row. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

**Returns**

The stride in bytes.

Implements [Image](#).

**10.100.3.36** `uint64_t GetTimeStamp ( ) const [virtual]`

Gets the time stamp for the image in nanoseconds.

**Returns**

The time stamp of the image.

Implements [Image](#).

**10.100.3.37** `PayloadTypeInfoIds GetTLPayloadType ( ) const [virtual]`

Gets the GenTL specific payload type that was transmitted.

This is a Transport Layer specific value that identifies how the image was transmitted. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

**Returns**

Transport Layer specific payload type.

Implements [Image](#).



**10.100.3.38** `uint64_t GetTLPixelFormat ( ) const [virtual]`

Gets the pixel format of the image.

This is a Transport Layer specific pixel format that identifies how the pixels in the image should be interpreted. To understand how to interpret this value it is necessary to know what the transport layer namespace is. This can be retrieved through a call to [GetTLPixelFormatNamespace\(\)](#). This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

See also

[GetTLPixelFormatNamespace\(\)](#)

Returns

Transport Layer specific pixel format.

Implements [IImage](#).

**10.100.3.39** `PixelFormatNamespaceID GetTLPixelFormatNamespace ( ) const [virtual]`

Returns an enum value that represents the namespace in which this image's TL specific pixel format resides.

This information is important to properly interpret the value returned by [GetTLPixelFormat\(\)](#)

See also

[GetTLPixelFormat\(\)](#)

Returns

enum value representing the PixelFormatNamespace.

Implements [IImage](#).

**10.100.3.40** `size_t GetValidPayloadSize ( ) const [virtual]`

Returns the size of valid data in the image payload.

This is the actual amount of data read from the device. A user created image has a payload size of zero. [Get↔BufferSize\(\)](#) returns the total size of bytes allocated for the image.

See also

[GetBufferSize\(\)](#)

Returns

size\_t value representing valid payload.

Implements [IImage](#).

**10.100.3.41** `size_t GetWidth ( ) const [virtual]`

Gets the width of the image in pixels.

This information is retrieved from the Transport Layer image format headers. It is retrieved on a per image basis.

**Returns**

The width in pixels.

Implements [IImage](#).

**10.100.3.42** `size_t GetXOffset ( ) const [virtual]`

Gets the ROI x offset in pixels for this image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

**Returns**

The x offset in pixels.

Implements [IImage](#).

**10.100.3.43** `size_t GetXPadding ( ) const [virtual]`

Gets the x padding in bytes for this image.

This is the number of bytes at the end of each line to facilitate alignment in buffers. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

**Returns**

The x padding in bytes.

Implements [IImage](#).

**10.100.3.44** `size_t GetYOffset ( ) const [virtual]`

Gets the ROI y offset in pixels for this image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

**Returns**

The y offset in pixels.

Implements [IImage](#).

**10.100.3.45** `size_t GetYPadding ( ) const [virtual]`

Gets the y padding in bytes for this image.

This is the number of bytes at the end of each image to facilitate alignment in buffers. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

**Returns**

The y padding in bytes.

Implements [IImage](#).

**10.100.3.46** `bool HasCRC ( ) const [virtual]`

Checks if the image contains ImageCRC checksum from chunk data.

**Returns**

Returns true if image contains ImageCRC checksum from chunk data and false otherwise.

Implements [IImage](#).

**10.100.3.47** `bool IsIncomplete ( ) const [virtual]`

Returns a boolean value indicating if this image was incomplete.

An image is marked as incomplete if the transport layer received less data then it requested.

**Returns**

Returns true if image is incomplete, false otherwise.

Implements [IImage](#).

**10.100.3.48** `bool IsInUse ( ) [virtual]`

Returns true if the image is still in use by the stream.

**Returns**

Returns true if the image is in use and false otherwise.

Implements [IImage](#).

**10.100.3.49** `void Release ( ) [virtual]`

Implements [IImage](#).

**10.100.3.50** `void ResetImage ( size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat ) [virtual]`

Sets new dimensions of the image object and allocates memory.

## Parameters

|                    |                                       |
|--------------------|---------------------------------------|
| <i>width</i>       | The width of image in pixels to set.  |
| <i>height</i>      | The height of image in pixels to set. |
| <i>offsetX</i>     | The x offset in pixels to set.        |
| <i>offsetY</i>     | The y offset in pixels to set.        |
| <i>pixelFormat</i> | Pixel format to set.                  |

Implements [IImage](#).

10.100.3.51 `void ResetImage ( size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void * pData ) [virtual]`

Sets new dimensions of the image object.

## Parameters

|                    |                                       |
|--------------------|---------------------------------------|
| <i>width</i>       | The width of image in pixels to set.  |
| <i>height</i>      | The height of image in pixels to set. |
| <i>offsetX</i>     | The x offset in pixels to set.        |
| <i>offsetY</i>     | The y offset in pixels to set.        |
| <i>pixelFormat</i> | Pixel format to set.                  |
| <i>pData</i>       | Pointer to the image buffer.          |

Implements [IImage](#).

10.100.3.52 `void Save ( const char * pFilename, ImageFileFormat format = FROM_FILE_EXT ) [virtual]`

Saves the image to the specified file name with the file format specified.

## Parameters

|                  |                              |
|------------------|------------------------------|
| <i>pFilename</i> | Filename to save image with. |
| <i>format</i>    | File format to save in.      |

Implements [IImage](#).

10.100.3.53 `void Save ( const char * pFilename, PNGOption & pOption ) [virtual]`

Saves the image to the specified file name with the options specified.

## Parameters

|                  |                                    |
|------------------|------------------------------------|
| <i>pFilename</i> | Filename to save image with.       |
| <i>pOption</i>   | Options to use while saving image. |

Implements [IImage](#).

10.100.3.54 `void Save ( const char * pFilename, PPMOption & pOption )` [virtual]

Saves the image to the specified file name with the options specified.

Parameters

|                  |                                    |
|------------------|------------------------------------|
| <i>pFilename</i> | Filename to save image with.       |
| <i>pOption</i>   | Options to use while saving image. |

Implements [IImage](#).

10.100.3.55 `void Save ( const char * pFilename, PGMOption & pOption )` [virtual]

Saves the image to the specified file name with the options specified.

Parameters

|                  |                                    |
|------------------|------------------------------------|
| <i>pFilename</i> | Filename to save image with.       |
| <i>pOption</i>   | Options to use while saving image. |

Implements [IImage](#).

10.100.3.56 `void Save ( const char * pFilename, TIFFOption & pOption )` [virtual]

Saves the image to the specified file name with the options specified.

Parameters

|                  |                                    |
|------------------|------------------------------------|
| <i>pFilename</i> | Filename to save image with.       |
| <i>pOption</i>   | Options to use while saving image. |

Implements [IImage](#).

10.100.3.57 `void Save ( const char * pFilename, JPEGOption & pOption )` [virtual]

Saves the image to the specified file name with the options specified.

Parameters

|                  |                                    |
|------------------|------------------------------------|
| <i>pFilename</i> | Filename to save image with.       |
| <i>pOption</i>   | Options to use while saving image. |

Implements [IImage](#).

10.100.3.58 **void Save ( const char \* *pFilename*, **JPG2Option** & *pOption* )** [virtual]

Saves the image to the specified file name with the options specified.

#### Parameters

|                  |                                    |
|------------------|------------------------------------|
| <i>pFilename</i> | Filename to save image with.       |
| <i>pOption</i>   | Options to use while saving image. |

Implements [IImage](#).

10.100.3.59 **void Save ( const char \* *pFilename*, **BMPOption** & *pOption* )** [virtual]

Saves the image to the specified file name with the options specified.

#### Parameters

|                  |                                    |
|------------------|------------------------------------|
| <i>pFilename</i> | Filename to save image with.       |
| <i>pOption</i>   | Options to use while saving image. |

Implements [IImage](#).

10.100.3.60 **static void SetDefaultColorProcessing ( **ColorProcessingAlgorithm** *colorAlgorithm* )** [static]

Sets the default color processing algorithm.

This method will be used for any image with the DEFAULT algorithm set. The method used is determined at the time of the [Convert\(\)](#) call, therefore the most recent execution of this function will take precedence. The default setting is shared within the current process.

#### Parameters

|                       |                                        |
|-----------------------|----------------------------------------|
| <i>colorAlgorithm</i> | The color processing algorithm to set. |
|-----------------------|----------------------------------------|

#### See also

[GetDefaultColorProcessing\(\)](#)

10.100.3.61 **static void SetHeatMapColorGradient ( const **HeatMapColor** *newLowColor*, const **HeatMapColor** *newHighColor* )** [static]

Sets the heatmap gradient color vector to the new desired range between HEATMAP\_BLACK and HEATMAP\_↔ WHITE.

Heatmap images are currently implemented for polarized cameras only.

## Parameters

|                     |                                           |
|---------------------|-------------------------------------------|
| <i>newLowColor</i>  | New color at which to begin the gradient. |
| <i>newHighColor</i> | New color at which to end the gradient.   |

**10.100.3.62** `static void SetHeatMapRange ( const unsigned int newLowValue, const unsigned int newHighValue )`  
`[static]`

Sets the high and low values used to determine which grayscale values are converted to a color 'heatmap' representation.

Acceptable values range from 0 to 100. Heatmap images are currently implemented for polarized cameras only.

## Parameters

|                     |                                                   |
|---------------------|---------------------------------------------------|
| <i>newLowValue</i>  | New value at which to begin color representation. |
| <i>newHighValue</i> | New value at which to end color representation.   |

**10.100.4 Friends And Related Function Documentation**

**10.100.4.1** `friend class IDataStream` `[friend]`

**10.100.4.2** `friend class ImageConverter` `[friend]`

**10.100.4.3** `friend class ImageFiler` `[friend]`

**10.100.4.4** `friend class ImageStatsCalculator` `[friend]`

**10.100.4.5** `friend class Stream` `[friend]`

**10.100.5 Member Data Documentation**

**10.100.5.1** `ImageData* m_pImageData` `[protected]`

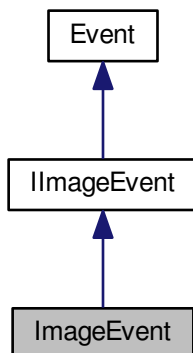
The documentation for this class was generated from the following file:

- [include/Image.h](#)

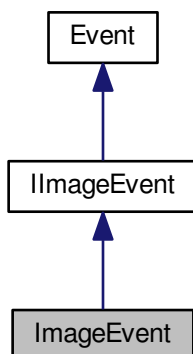
## 10.101 ImageEvent Class Reference

A handler for capturing image arrival events.

Inheritance diagram for ImageEvent:



Collaboration diagram for ImageEvent:



### Public Member Functions

- [ImageEvent](#) ()  
*Default Constructor.*
- virtual [~ImageEvent](#) ()  
*Virtual Destructor.*
- virtual void [OnImageEvent](#) ([ImagePtr](#) image)=0  
*[Image](#) event callback.*



## Protected Member Functions

- [ImageEvent](#) & `operator=` (const [ImageEvent](#) &)  
*Assignment operator.*

## Additional Inherited Members

### 10.101.1 Detailed Description

A handler for capturing image arrival events.

### 10.101.2 Constructor & Destructor Documentation

#### 10.101.2.1 ImageEvent ( )

Default Constructor.

#### 10.101.2.2 virtual ~ImageEvent ( ) [virtual]

Virtual Destructor.

### 10.101.3 Member Function Documentation

#### 10.101.3.1 virtual void OnImageEvent ( ImagePtr image ) [pure virtual]

[Image](#) event callback.

##### Parameters

|              |                                     |
|--------------|-------------------------------------|
| <i>image</i> | The <a href="#">ImagePtr</a> object |
|--------------|-------------------------------------|

Implements [ImageEvent](#).

#### 10.101.3.2 ImageEvent& operator= ( const ImageEvent & ) [protected]

Assignment operator.

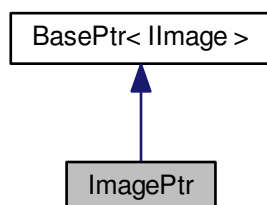
The documentation for this class was generated from the following file:

- include/[ImageEvent.h](#)

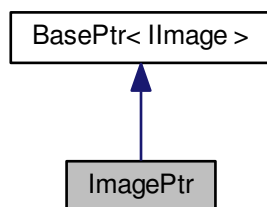
## 10.102 ImagePtr Class Reference

A reference tracked pointer to an image object.

Inheritance diagram for ImagePtr:



Collaboration diagram for ImagePtr:



### Public Member Functions

- [ImagePtr](#) ()  
*Default constructor.*
- [ImagePtr](#) (const int)  
*Default constructor.*
- virtual [~ImagePtr](#) (void)  
*Virtual destructor.*
- virtual [ImagePtr](#) & [operator=](#) (const [ImagePtr](#) &)  
*Assignment operator.*
- virtual [ImagePtr](#) & [operator=](#) (const int nMustBeNull)  
*Assignment operator.*

## Additional Inherited Members

### 10.102.1 Detailed Description

A reference tracked pointer to an image object.

When the [ImagePtr](#) goes out of scope, it will trigger an auto release of the image from the stream.

### 10.102.2 Constructor & Destructor Documentation

#### 10.102.2.1 ImagePtr ( )

Default constructor.

#### 10.102.2.2 ImagePtr ( const int )

Default constructor.

#### 10.102.2.3 virtual ~ImagePtr ( void ) [virtual]

Virtual destructor.

### 10.102.3 Member Function Documentation

#### 10.102.3.1 virtual ImagePtr& operator= ( const ImagePtr & ) [virtual]

Assignment operator.

#### 10.102.3.2 virtual ImagePtr& operator= ( const int *nMustBeNull* ) [virtual]

Assignment operator.

Reimplemented from [BasePtr< Image >](#).

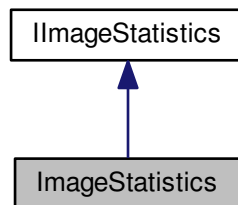
The documentation for this class was generated from the following file:

- [include/ImagePtr.h](#)

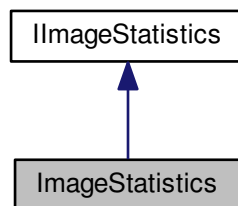
## 10.103 ImageStatistics Class Reference

Represents image statistics for an image.

Inheritance diagram for ImageStatistics:



Collaboration diagram for ImageStatistics:



### Public Member Functions

- [ImageStatistics](#) ()  
*Default constructor.*
- virtual [~ImageStatistics](#) ()  
*Default destructor.*
- [ImageStatistics](#) (const [ImageStatistics](#) &other)  
*Copy constructor.*
- [ImageStatistics](#) & [operator=](#) (const [ImageStatistics](#) &other)  
*Assignment operator.*
- virtual void [EnableAll](#) ()  
*Enable all channels.*
- virtual void [DisableAll](#) ()  
*Disable all channels.*
- virtual void [EnableGreyOnly](#) ()

- Enable only the grey channel.*

  - virtual void [EnableRGBOOnly](#) ()

*Enable only the RGB channels.*

  - virtual void [EnableHSLOnly](#) ()

*Enable only the HSL channels.*

  - virtual void [GetChannelStatus](#) ([StatisticsChannel](#) channel, bool \*pEnabled) const

*Gets the status of a statistics channel.*

  - virtual void [SetChannelStatus](#) ([StatisticsChannel](#) channel, bool enabled)

*Sets the status of a statistics channel.*

  - virtual void [GetRange](#) ([StatisticsChannel](#) channel, unsigned int \*pMin, unsigned int \*pMax) const

*Gets the range of a statistics channel.*

  - virtual void [GetPixelValueRange](#) ([StatisticsChannel](#) channel, unsigned int \*pPixelValueMin, unsigned int \*pPixelValueMax) const

*Gets the range of a statistics channel.*

  - virtual void [GetNumPixelValues](#) ([StatisticsChannel](#) channel, unsigned int \*pNumPixelValues) const

*Gets the number of unique pixel values in the image.*

  - virtual void [GetMean](#) ([StatisticsChannel](#) channel, float \*pPixelValueMean) const

*Gets the mean of the image.*

  - virtual void [GetHistogram](#) ([StatisticsChannel](#) channel, int \*\*ppHistogram) const

*Gets the histogram for the image.*

  - virtual void [GetStatistics](#) ([StatisticsChannel](#) channel, unsigned int \*pRangeMin=NULL, unsigned int \*pRangeMax=NULL, unsigned int \*pPixelValueMin=NULL, unsigned int \*pPixelValueMax=NULL, unsigned int \*pNumPixelValues=NULL, float \*pPixelValueMean=NULL, int \*\*ppHistogram=NULL) const

*Gets all statistics for the image.*

## Friends

- class [ImageStatsCalculator](#)

## Additional Inherited Members

### 10.103.1 Detailed Description

Represents image statistics for an image.

### 10.103.2 Constructor & Destructor Documentation

#### 10.103.2.1 ImageStatistics ( )

Default constructor.

#### 10.103.2.2 virtual ~ImageStatistics ( ) [virtual]

Default destructor.

### 10.103.2.3 ImageStatistics ( const ImageStatistics & other )

Copy constructor.

## 10.103.3 Member Function Documentation

### 10.103.3.1 virtual void DisableAll ( ) [virtual]

Disable all channels.

Implements [IImageStatistics](#).

### 10.103.3.2 virtual void EnableAll ( ) [virtual]

Enable all channels.

Implements [IImageStatistics](#).

### 10.103.3.3 virtual void EnableGreyOnly ( ) [virtual]

Enable only the grey channel.

Implements [IImageStatistics](#).

### 10.103.3.4 virtual void EnableHSLOnly ( ) [virtual]

Enable only the HSL channels.

Implements [IImageStatistics](#).

### 10.103.3.5 virtual void EnableRGBOnly ( ) [virtual]

Enable only the RGB channels.

Implements [IImageStatistics](#).

### 10.103.3.6 virtual void GetChannelStatus ( StatisticsChannel channel, bool \* pEnabled ) const [virtual]

Gets the status of a statistics channel.

#### Parameters

|                 |                                 |
|-----------------|---------------------------------|
| <i>channel</i>  | The statistics channel.         |
| <i>pEnabled</i> | Whether the channel is enabled. |

See also

[SetChannelStatus\(\)](#)

Implements [IImageStatistics](#).

10.103.3.7 virtual void GetHistogram ( **StatisticsChannel** *channel*, int \*\* *ppHistogram* ) const [virtual]

Gets the histogram for the image.

Parameters

|                    |                                               |
|--------------------|-----------------------------------------------|
| <i>channel</i>     | The statistics channel.                       |
| <i>ppHistogram</i> | Pointer to an array containing the histogram. |

Implements [IImageStatistics](#).

10.103.3.8 virtual void GetMean ( **StatisticsChannel** *channel*, float \* *pPixelValueMean* ) const [virtual]

Gets the mean of the image.

Parameters

|                        |                         |
|------------------------|-------------------------|
| <i>channel</i>         | The statistics channel. |
| <i>pPixelValueMean</i> | The mean of the image.  |

Implements [IImageStatistics](#).

10.103.3.9 virtual void GetNumPixelValues ( **StatisticsChannel** *channel*, unsigned int \* *pNumPixelValues* ) const [virtual]

Gets the number of unique pixel values in the image.

Parameters

|                        |                                    |
|------------------------|------------------------------------|
| <i>channel</i>         | The statistics channel.            |
| <i>pNumPixelValues</i> | The number of unique pixel values. |

Implements [IImageStatistics](#).

10.103.3.10 virtual void GetPixelValueRange ( **StatisticsChannel** *channel*, unsigned int \* *pPixelValueMin*, unsigned int \* *pPixelValueMax* ) const [virtual]

Gets the range of a statistics channel.

The values returned are the maximum values recorded for all pixels in the image.

## Parameters

|                       |                          |
|-----------------------|--------------------------|
| <i>channel</i>        | The statistics channel.  |
| <i>pPixelValueMin</i> | The minimum pixel value. |
| <i>pPixelValueMax</i> | The maximum pixel value. |

Implements [IImageStatistics](#).

10.103.3.11 `virtual void GetRange ( StatisticsChannel channel, unsigned int * pMin, unsigned int * pMax ) const`  
[virtual]

Gets the range of a statistics channel.

The values returned are the maximum possible values for any given pixel in the image. This is generally 0-255 for 8 bit images, and 0-65535 for 16 bit images.

## Parameters

|                |                             |
|----------------|-----------------------------|
| <i>channel</i> | The statistics channel.     |
| <i>pMin</i>    | The minimum possible value. |
| <i>pMax</i>    | The maximum possible value. |

Implements [IImageStatistics](#).

10.103.3.12 `virtual void GetStatistics ( StatisticsChannel channel, unsigned int * pRangeMin = NULL, unsigned int * pRangeMax = NULL, unsigned int * pPixelValueMin = NULL, unsigned int * pPixelValueMax = NULL, unsigned int * pNumPixelValues = NULL, float * pPixelValueMean = NULL, int ** ppHistogram = NULL ) const`  
[virtual]

Gets all statistics for the image.

## Parameters

|                        |                                               |
|------------------------|-----------------------------------------------|
| <i>channel</i>         | The statistics channel.                       |
| <i>pRangeMin</i>       | The minimum possible value.                   |
| <i>pRangeMax</i>       | The maximum possible value.                   |
| <i>pPixelValueMin</i>  | The minimum pixel value.                      |
| <i>pPixelValueMax</i>  | The maximum pixel value.                      |
| <i>pNumPixelValues</i> | The number of unique pixel values.            |
| <i>pPixelValueMean</i> | The mean of the image.                        |
| <i>ppHistogram</i>     | Pointer to an array containing the histogram. |

Implements [IImageStatistics](#).

10.103.3.13 `ImageStatistics& operator= ( const ImageStatistics & other )`

Assignment operator.



## Parameters

|              |                                                          |
|--------------|----------------------------------------------------------|
| <i>other</i> | The <a href="#">ImageStatistics</a> object to copy from. |
|--------------|----------------------------------------------------------|

10.103.3.14 `virtual void SetChannelStatus ( StatisticsChannel channel, bool enabled )` `[virtual]`

Sets the status of a statistics channel.

## Parameters

|                |                                        |
|----------------|----------------------------------------|
| <i>channel</i> | The statistics channel.                |
| <i>enabled</i> | Whether the channel should be enabled. |

## See also

[GetChannelStatus\(\)](#)

Implements [IImageStatistics](#).

## 10.103.4 Friends And Related Function Documentation

10.103.4.1 `friend class ImageStatsCalculator` `[friend]`

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

- [include/ImageStatistics.h](#)

## 10.104 int64\_autovector\_t Class Reference

Vector of integers with reference counting.

## Public Member Functions

- [int64\\_autovector\\_t](#) ()
- [int64\\_autovector\\_t](#) (const [int64\\_autovector\\_t](#) &obj)
- [int64\\_autovector\\_t](#) (size\_t n)
- virtual [~int64\\_autovector\\_t](#) (void)
- [int64\\_autovector\\_t](#) & [operator=](#) (const [int64\\_autovector\\_t](#) &obj)
- void [operator delete](#) (void \*pWhere)
- void \* [operator new](#) (size\_t uiSize)
- [int64\\_t](#) & [operator\[\]](#) (size\_t uiIndex)
- const [int64\\_t](#) & [operator\[\]](#) (size\_t uiIndex) const
- size\_t [size](#) () const

## Protected Attributes

- `std::vector< int64_t > * _pv`
- `ATOMIC_VARIABLE * _pCount`

### 10.104.1 Detailed Description

Vector of integers with reference counting.

### 10.104.2 Constructor & Destructor Documentation

10.104.2.1 `int64_autovector_t ( )`

10.104.2.2 `int64_autovector_t ( const int64_autovector_t & obj )`

10.104.2.3 `int64_autovector_t ( size_t n ) [explicit]`

10.104.2.4 `virtual ~int64_autovector_t ( void ) [virtual]`

### 10.104.3 Member Function Documentation

10.104.3.1 `void operator delete ( void * pWhere )`

10.104.3.2 `void* operator new ( size_t uiSize )`

10.104.3.3 `int64_autovector_t& operator= ( const int64_autovector_t & obj )`

10.104.3.4 `int64_t& operator[] ( size_t uiIndex )`

10.104.3.5 `const int64_t& operator[] ( size_t uiIndex ) const`

10.104.3.6 `size_t size ( ) const`

### 10.104.4 Member Data Documentation

10.104.4.1 `ATOMIC_VARIABLE* _pCount [protected]`

10.104.4.2 `std::vector<int64_t>* _pv [protected]`

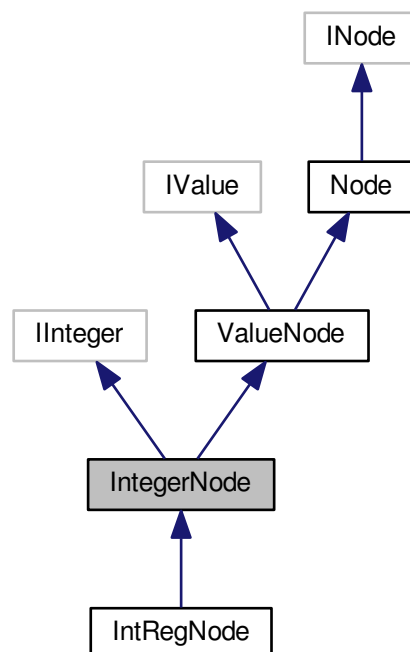
The documentation for this class was generated from the following file:

- `include/SpinGenApi/Autovector.h`

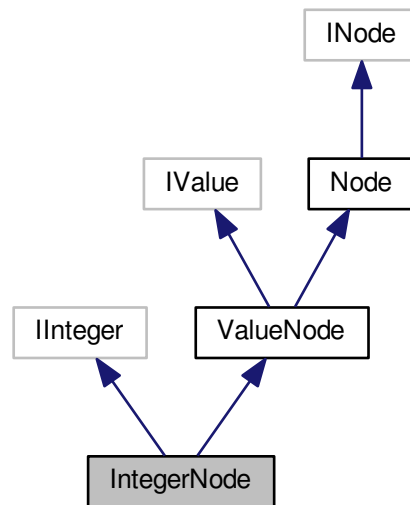
## 10.105 IntegerNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for IntegerNode:



Collaboration diagram for IntegerNode:



## Public Member Functions

- [IntegerNode](#) ()
- [IntegerNode](#) (std::shared\_ptr< Node::NodeImpl > pInteger)
- virtual [~IntegerNode](#) ()
- virtual void [SetValue](#) (int64\_t Value, bool [Verify](#)=true)
  - Set node value.*
- virtual [Integer](#) & [operator=](#) (int64\_t Value)
  - Set node value.*
- virtual int64\_t [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
  - Get node value.*
- virtual int64\_t [operator\(\)](#) ()
  - Get node value.*
- virtual int64\_t [operator\\*](#) ()
  - Get node value.*
- virtual int64\_t [GetMin](#) ()
  - Get minimum value allowed.*
- virtual int64\_t [GetMax](#) ()
  - Get maximum value allowed.*
- virtual [EIncMode](#) [GetIncMode](#) ()
  - Get increment mode.*
- virtual int64\_t [GetInc](#) ()
  - Get increment.*
- virtual [int64\\_autovector\\_t](#) [GetListOfValidValues](#) (bool bounded=true)
  - Get list of valid value.*
- virtual [ERepresentation](#) [GetRepresentation](#) ()
  - Get recommended representation.*

- virtual [GenlCam::gcstring GetUnit](#) ()  
*Get the physical unit name.*
- virtual [IFloat \\* GetFloatAlias](#) ()  
*gets the interface of an alias node.*
- virtual void [ImposeMin](#) (int64\_t Value)  
*Restrict minimum value.*
- virtual void [ImposeMax](#) (int64\_t Value)  
*Restrict maximum value.*
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Integer*

## Additional Inherited Members

### 10.105.1 Detailed Description

[Interface](#) for string properties.

### 10.105.2 Constructor & Destructor Documentation

10.105.2.1 [IntegerNode](#) ( )

10.105.2.2 [IntegerNode](#) ( std::shared\_ptr< Node::NodeImpl > pInteger )

10.105.2.3 virtual [~IntegerNode](#) ( ) [virtual]

### 10.105.3 Member Function Documentation

10.105.3.1 virtual [IFloat\\* GetFloatAlias](#) ( ) [virtual]

gets the interface of an alias node.

10.105.3.2 virtual [int64\\_t GetInc](#) ( ) [virtual]

Get increment.

10.105.3.3 virtual [EIncMode GetIncMode](#) ( ) [virtual]

Get increment mode.

10.105.3.4 virtual [int64\\_autovector\\_t GetListOfValidValues](#) ( bool bounded = true ) [virtual]

Get list of valid value.

10.105.3.5 virtual int64\_t GetMax ( ) [virtual]

Get maximum value allowed.

10.105.3.6 virtual int64\_t GetMin ( ) [virtual]

Get minimum value allowed.

10.105.3.7 virtual ERepresentation GetRepresentation ( ) [virtual]

Get recommended representation.

10.105.3.8 virtual GenICam::gcstring GetUnit ( ) [virtual]

Get the physical unit name.

10.105.3.9 virtual int64\_t GetValue ( bool *Verify* = false, bool *IgnoreCache* = false ) [virtual]

Get node value.

#### Parameters

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| <i>Verify</i>      | Enables Range verification (default = false). The AccessMode is always checked |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false)                |

#### Returns

The value read

10.105.3.10 virtual void ImposeMax ( int64\_t *Value* ) [virtual]

Restrict maximum value.

10.105.3.11 virtual void ImposeMin ( int64\_t *Value* ) [virtual]

Restrict minimum value.

10.105.3.12 virtual int64\_t operator()( ) [virtual]

Get node value.

10.105.3.13 `virtual int64_t operator*( ) [virtual]`

Get node value.

10.105.3.14 `virtual Integer& operator=( int64_t Value ) [virtual]`

Set node value.

10.105.3.15 `virtual void SetReference ( INode * pBase ) [virtual]`

overload SetReference for Integer

Reimplemented from [ValueNode](#).

Reimplemented in [IntRegNode](#).

10.105.3.16 `virtual void SetValue ( int64_t Value, bool Verify = true ) [virtual]`

Set node value.

#### Parameters

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>Value</i>  | The value to set                                           |
| <i>Verify</i> | Enables AccessMode and Range verification (default = true) |

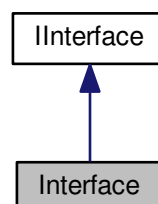
The documentation for this class was generated from the following file:

- [include/SpinGenApi/IntegerNode.h](#)

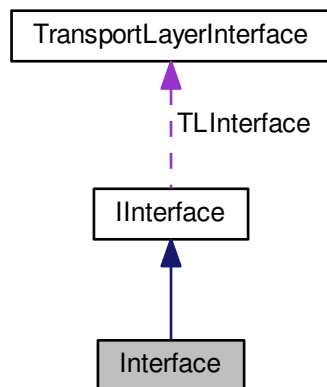
## 10.106 Interface Class Reference

An interface object which holds a list of cameras.

Inheritance diagram for Interface:



Collaboration diagram for Interface:



## Public Member Functions

- virtual [~Interface](#) (void)  
*Virtual Destructor.*
- [CameraList GetCameras](#) (bool updateCameras=true) const  
*Returns a list of cameras available on this interface.*
- bool [UpdateCameras](#) ()  
*Updates the list of cameras on this interface.*
- [GenApi::INodeMap & GetTLNodeMap](#) () const  
*Gets a nodeMap that is generated from a [GenICam](#) XML file for the GenTL interface Module.*
- void [RegisterEvent](#) ([Event](#) &evtToRegister)  
*Registers an event for the interface.*
- void [UnregisterEvent](#) ([Event](#) &evtToUnregister)  
*Unregisters an event for the interface.*
- bool [IsInUse](#) () const  
*Checks if the interface is in use by any camera objects.*
- void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int \*pResultSize=0, [ActionCommandResult](#) results[]=NULL) const  
*Broadcast an Action Command to all devices on interface.*

## Friends

- class [InterfaceInternal](#)

## Additional Inherited Members

### 10.106.1 Detailed Description

An interface object which holds a list of cameras.



## 10.106.2 Constructor & Destructor Documentation

10.106.2.1 `virtual ~Interface ( void ) [virtual]`

Virtual Destructor.

## 10.106.3 Member Function Documentation

10.106.3.1 `CameraList GetCameras ( bool updateCameras = true ) const [virtual]`

Returns a list of cameras available on this interface.

This call returns either usb3 vision or gige vision cameras depending on the underlying transport layer of this interface. The camera list object will reference count the cameras that it holds. It is important that the [CameraList](#) is destroyed or is cleared before [System::ReleaseInstance\(\)](#) can be called or an [InterfaceList](#) that holds this interface can be cleared.

See also

[System::ReleaseInstance\(\)](#)  
[InterfaceList::Clear\(\)](#)  
[CameraList::Clear\(\)](#)

Parameters

|                      |                                                                                                     |
|----------------------|-----------------------------------------------------------------------------------------------------|
| <i>updateCameras</i> | A flag used to issue an <code>updateCameras()</code> call internally before getting the camera list |
|----------------------|-----------------------------------------------------------------------------------------------------|

Returns

An [CameraList](#) object that contains a list of cameras on this interface.

Implements [Interface](#).

10.106.3.2 `GenApi::INodeMap& GetTLNodeMap ( ) const [virtual]`

Gets a nodeMap that is generated from a [GenICam](#) XML file for the GenTL interface Module.

Returns

A reference to a `INodeMap` object.

Implements [Interface](#).

10.106.3.3 `bool IsInUse ( ) const [virtual]`

Checks if the interface is in use by any camera objects.

Returns

Returns true if the interface is in use and false otherwise.

Implements [Interface](#).

10.106.3.4 void RegisterEvent ( Event & *evtToRegister* ) [virtual]

Registers an event for the interface.

#### Parameters

|                      |                                         |
|----------------------|-----------------------------------------|
| <i>evtToRegister</i> | The event to register for the interface |
|----------------------|-----------------------------------------|

Implements [Interface](#).

10.106.3.5 void SendActionCommand ( unsigned int *deviceKey*, unsigned int *groupKey*, unsigned int *groupMask*, unsigned long long *actionTime* = 0, unsigned int \* *pResultSize* = 0, ActionCommandResult *results*[] = NULL ) const [virtual]

Broadcast an Action Command to all devices on interface.

#### Parameters

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>deviceKey</i>   | The Action Command's device key                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>groupKey</i>    | The Action Command's group key                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <i>groupMask</i>   | The Action Command's group mask                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>actionTime</i>  | (Optional) Time when to assert a future action. Zero means immediate action.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>pResultSize</i> | (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results.                                                                                                                                                                                                                                                                                                                                |
| <i>results</i>     | (Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL. |

Implements [Interface](#).

10.106.3.6 void UnregisterEvent ( Event & *evtToUnregister* ) [virtual]

Unregisters an event for the interface.

#### Parameters

|                        |                                            |
|------------------------|--------------------------------------------|
| <i>evtToUnregister</i> | The event to unregister from the interface |
|------------------------|--------------------------------------------|

Implements [Interface](#).

10.106.3.7 bool UpdateCameras ( ) [virtual]

Updates the list of cameras on this interface.

This function needs to be called before any cameras can be discovered using [GetCameras\(\)](#). [System::GetCameras\(\)](#) will automatically call this function for each interface it enumerates. If the list changed after the last time [System::GetCameras\(\)](#) or [UpdateCameras\(\)](#) was called then the return value will be true, otherwise it is false.

See also

[System::GetCameras\(\)](#)  
[GetCameras\(\)](#)

Returns

true if cameras changed on interface and false otherwise.

Implements [IInterface](#).

## 10.106.4 Friends And Related Function Documentation

10.106.4.1 `friend class InterfaceInternal` `[friend]`

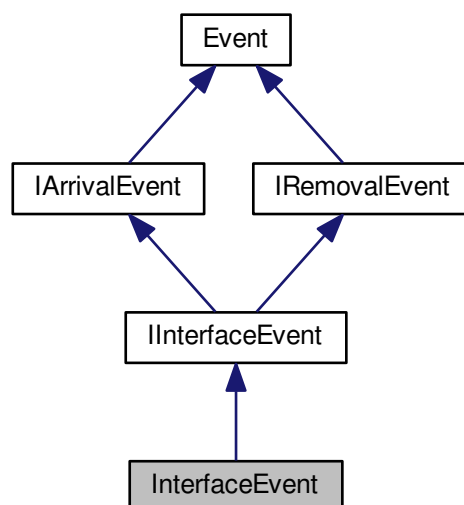
The documentation for this class was generated from the following file:

- `include/Interface.h`

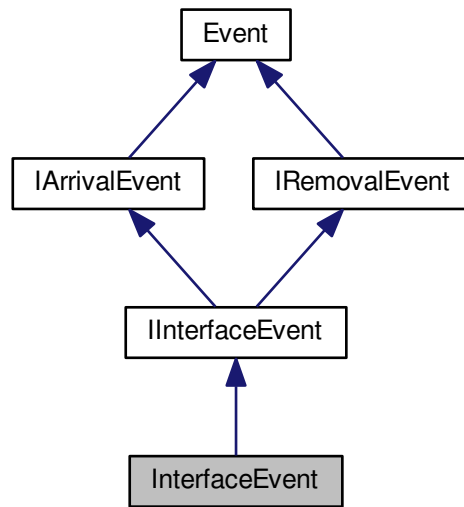
## 10.107 InterfaceEvent Class Reference

A handler to device arrival and removal events on all interfaces.

Inheritance diagram for InterfaceEvent:



Collaboration diagram for InterfaceEvent:



## Public Member Functions

- [InterfaceEvent](#) ()  
*Default constructor.*
- virtual [~InterfaceEvent](#) ()  
*Virtual destructor.*
- virtual void [OnDeviceArrival](#) (uint64\_t serialNumber)=0  
*Device arrival event callback.*
- virtual void [OnDeviceRemoval](#) (uint64\_t serialNumber)=0  
*Callback to the device removal event.*

## Protected Member Functions

- [InterfaceEvent](#) & [operator=](#) (const [InterfaceEvent](#) &)  
*Assignment operator.*

## Additional Inherited Members

### 10.107.1 Detailed Description

A handler to device arrival and removal events on all interfaces.

## 10.107.2 Constructor & Destructor Documentation

### 10.107.2.1 InterfaceEvent ( )

Default constructor.

### 10.107.2.2 virtual ~InterfaceEvent ( ) [virtual]

Virtual destructor.

## 10.107.3 Member Function Documentation

### 10.107.3.1 virtual void OnDeviceArrival ( uint64\_t *serialNumber* ) [pure virtual]

Device arrival event callback.

Implements [IInterfaceEvent](#).

### 10.107.3.2 virtual void OnDeviceRemoval ( uint64\_t *serialNumber* ) [pure virtual]

Callback to the device removal event.

#### Parameters

|                     |                                         |
|---------------------|-----------------------------------------|
| <i>serialNumber</i> | The serial number of the removed device |
|---------------------|-----------------------------------------|

Implements [IInterfaceEvent](#).

### 10.107.3.3 InterfaceEvent& operator= ( const InterfaceEvent & ) [protected]

Assignment operator.

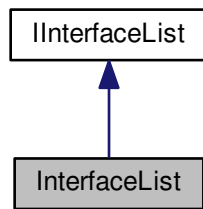
The documentation for this class was generated from the following file:

- [include/InterfaceEvent.h](#)

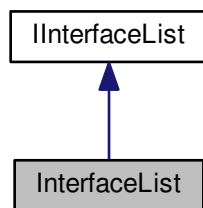
## 10.108 InterfaceList Class Reference

A list of the available interfaces on the system.

Inheritance diagram for InterfaceList:



Collaboration diagram for InterfaceList:



## Public Member Functions

- [InterfaceList](#) (void)
- virtual [~InterfaceList](#) (void)
- [InterfaceList](#) (const [InterfaceList](#) &iface)
- [InterfaceList](#) & [operator=](#) (const [InterfaceList](#) &iface)  
*Assignment operator.*
- [InterfacePtr](#) [operator\[\]](#) (unsigned int index)  
*Array subscription operators.*
- unsigned int [GetSize](#) () const  
*Returns the size of the interface list.*
- [InterfacePtr](#) [GetByIndex](#) (unsigned int index) const  
*Returns a pointer to an [Interface](#) object at the "index".*
- void [Clear](#) ()  
*Clears the list of interfaces and destroys their corresponding objects.*

## Friends

- class [SystemImpl](#)

## Additional Inherited Members

### 10.108.1 Detailed Description

A list of the available interfaces on the system.

### 10.108.2 Constructor & Destructor Documentation

10.108.2.1 `InterfaceList ( void )`

10.108.2.2 `virtual ~InterfaceList ( void ) [virtual]`

10.108.2.3 `InterfaceList ( const InterfaceList & iface )`

### 10.108.3 Member Function Documentation

10.108.3.1 `void Clear ( ) [virtual]`

Clears the list of interfaces and destroys their corresponding objects.

It is important to first make sure there are no referenced cameras still in use before calling [Clear\(\)](#). If a camera on any of the interfaces is still in use this function will throw an exception.

Implements [IInterfaceList](#).

10.108.3.2 `InterfacePtr GetByIndex ( unsigned int index ) const [virtual]`

Returns a pointer to an [Interface](#) object at the "index".

#### Parameters

|              |                                                                     |
|--------------|---------------------------------------------------------------------|
| <i>index</i> | The index at which to retrieve the <a href="#">Interface</a> object |
|--------------|---------------------------------------------------------------------|

#### Returns

A pointer to an [Interface](#) object.

Implements [IInterfaceList](#).

10.108.3.3 `unsigned int GetSize ( ) const [virtual]`

Returns the size of the interface list.

The size is the number of [Interface](#) objects stored in the list.

#### Returns

An integer that represents the list size.

Implements [IInterfaceList](#).

10.108.3.4 `InterfaceList& operator= ( const InterfaceList & iface )`

Assignment operator.

10.108.3.5 `InterfacePtr operator[] ( unsigned int index )` [virtual]

Array subscription operators.

Implements [InterfaceList](#).

## 10.108.4 Friends And Related Function Documentation

10.108.4.1 `friend class SystemImpl` [friend]

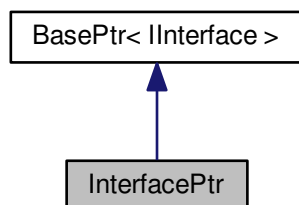
The documentation for this class was generated from the following file:

- include/[InterfaceList.h](#)

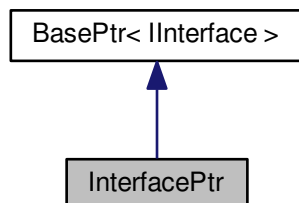
## 10.109 InterfacePtr Class Reference

A reference tracked pointer to the interface object.

Inheritance diagram for InterfacePtr:



Collaboration diagram for InterfacePtr:





## Public Member Functions

- [InterfacePtr](#) () throw ()  
*Default Constructor.*
- [InterfacePtr](#) (const int) throw ()  
*Default Constructor.*
- virtual [~InterfacePtr](#) (void)  
*Virtual Destructor.*
- virtual [InterfacePtr](#) & [operator=](#) (const int nMustBeNull)  
*Copy Constructor.*

## Additional Inherited Members

### 10.109.1 Detailed Description

A reference tracked pointer to the interface object.

### 10.109.2 Constructor & Destructor Documentation

#### 10.109.2.1 [InterfacePtr](#) ( ) throw ) [inline]

Default Constructor.

#### 10.109.2.2 [InterfacePtr](#) ( const int ) throw ) [inline]

Default Constructor.

#### 10.109.2.3 virtual [~InterfacePtr](#) ( void ) [inline],[virtual]

Virtual Destructor.

### 10.109.3 Member Function Documentation

#### 10.109.3.1 virtual [InterfacePtr](#)& [operator=](#) ( const int *nMustBeNull* ) [inline],[virtual]

Copy Constructor.

Reimplemented from [BasePtr< IInterface >](#).

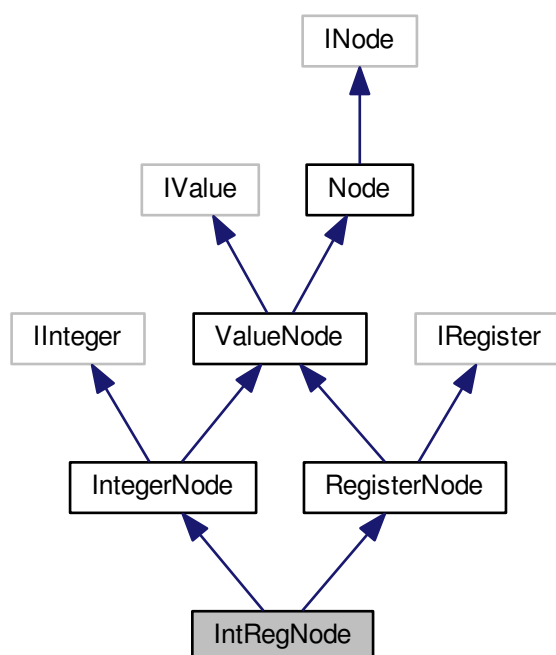
The documentation for this class was generated from the following file:

- include/[InterfacePtr.h](#)

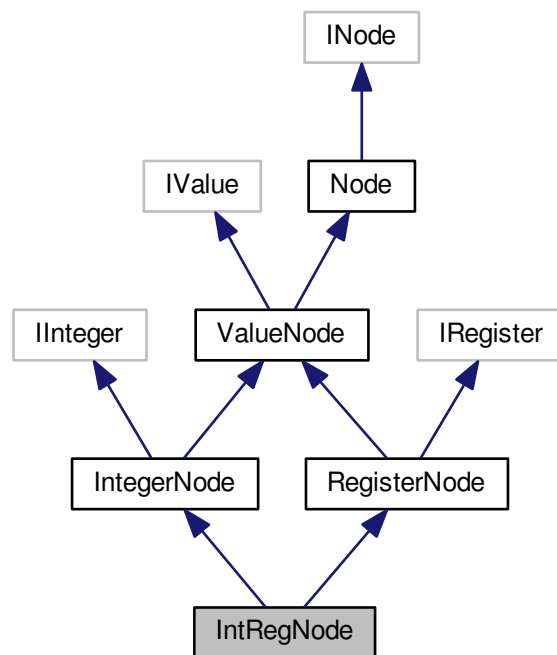
## 10.110 IntRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for IntRegNode:



Collaboration diagram for IntRegNode:



## Public Member Functions

- [IntRegNode](#) ()
- [IntRegNode](#) (std::shared\_ptr< Node::NodeImpl > pInteger)
- virtual [~IntRegNode](#) ()
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.110.1 Detailed Description

[Interface](#) for string properties.

### 10.110.2 Constructor & Destructor Documentation

#### 10.110.2.1 IntRegNode ( )

#### 10.110.2.2 IntRegNode ( std::shared\_ptr< Node::NodeImpl > pInteger )

10.110.2.3 `virtual ~IntRegNode ( ) [virtual]`

### 10.110.3 Member Function Documentation

10.110.3.1 `virtual void SetReference ( INode * pBase ) [virtual]`

overload SetReference for Value

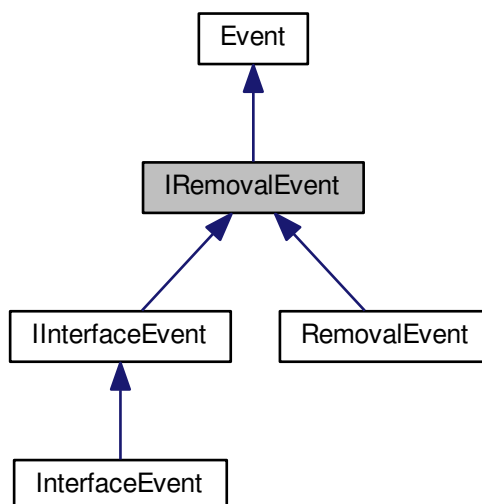
Reimplemented from [IntegerNode](#).

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

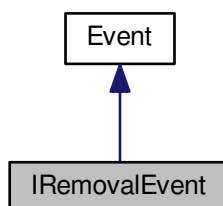
- `include/SpinGenApi/IntRegNode.h`

## 10.111 IRemovalEvent Class Reference

Inheritance diagram for IRemovalEvent:



Collaboration diagram for IRemovalEvent:



## Public Member Functions

- virtual [~IRemovalEvent](#) ()
- virtual void [OnDeviceRemoval](#) (uint64\_t serialNumber)=0

## Protected Member Functions

- [IRemovalEvent](#) ()
- [IRemovalEvent](#) (const [IRemovalEvent](#) &)
- [IRemovalEvent](#) & [operator=](#) (const [IRemovalEvent](#) &)

## Additional Inherited Members

### 10.111.1 Constructor & Destructor Documentation

10.111.1.1 virtual [~IRemovalEvent](#) ( ) [inline],[virtual]

10.111.1.2 [IRemovalEvent](#) ( ) [inline],[protected]

10.111.1.3 [IRemovalEvent](#) ( const [IRemovalEvent](#) & ) [inline],[protected]

### 10.111.2 Member Function Documentation

10.111.2.1 virtual void [OnDeviceRemoval](#) ( uint64\_t *serialNumber* ) [pure virtual]

Implemented in [InterfaceEvent](#), [RemovalEvent](#), and [IInterfaceEvent](#).

10.111.2.2 [IRemovalEvent& operator=](#) ( const [IRemovalEvent](#) & ) [protected]

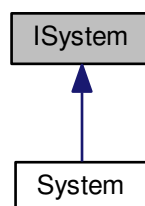
The documentation for this class was generated from the following file:

- include/Interface/[IRemovalEvent.h](#)

## 10.112 ISystem Class Reference

The interface file for [System](#).

Inheritance diagram for ISystem:



## Public Member Functions

- virtual [~ISystem](#) ()
- virtual void [ReleaseInstance](#) ()=0
- virtual [InterfaceList](#) [GetInterfaces](#) (bool *updateInterface*=true)=0
- virtual [CameraList](#) [GetCameras](#) (bool *updateInterfaces*=true, bool *updateCameras*=true)=0
- virtual bool [UpdateCameras](#) (bool *updateInterfaces*=true)=0
- virtual void [RegisterInterfaceEvent](#) ([Event](#) &evtToRegister, bool *updateInterface*=true)=0
- virtual void [UnregisterInterfaceEvent](#) ([Event](#) &evtToUnregister)=0
- virtual void [RegisterLoggingEvent](#) ([LoggingEvent](#) &handler)=0
- virtual void [UnregisterAllLoggingEvent](#) ()=0
- virtual void [UnregisterLoggingEvent](#) ([LoggingEvent](#) &handler)=0
- virtual void [SetLoggingEventPriorityLevel](#) ([SpinnakerLogLevel](#) *level*)=0
- virtual [SpinnakerLogLevel](#) [GetLoggingEventPriorityLevel](#) ()=0
- virtual bool [IsInUse](#) ()=0
- virtual void [SendActionCommand](#) (unsigned int *deviceKey*, unsigned int *groupKey*, unsigned int *groupMask*, unsigned long long *actionTime*=0, unsigned int \**pResultSize*=0, [ActionCommandResult](#) *results*[ ]=NULL)=0
- virtual const [LibraryVersion](#) [GetLibraryVersion](#) ()=0

## Protected Member Functions

- [ISystem](#) ()
- [ISystem](#) (const [ISystem](#) &)
- [ISystem](#) & [operator=](#) (const [ISystem](#) &)

### 10.112.1 Detailed Description

The interface file for [System](#).

### 10.112.2 Constructor & Destructor Documentation

10.112.2.1 virtual [~ISystem](#) ( ) [inline], [virtual]

10.112.2.2 [ISystem](#) ( ) [inline], [protected]

10.112.2.3 [ISystem](#) ( const [ISystem](#) & ) [inline], [protected]

### 10.112.3 Member Function Documentation

10.112.3.1 virtual [CameraList](#) [GetCameras](#) ( bool *updateInterfaces* = true, bool *updateCameras* = true ) [pure virtual]

Implemented in [System](#).

10.112.3.2 virtual [InterfaceList](#) [GetInterfaces](#) ( bool *updateInterface* = true ) [pure virtual]

Implemented in [System](#).

10.112.3.3 `virtual const LibraryVersion GetLibraryVersion ( ) [pure virtual]`

Implemented in [System](#).

10.112.3.4 `virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ( ) [pure virtual]`

Implemented in [System](#).

10.112.3.5 `virtual bool IsInUse ( ) [pure virtual]`

Implemented in [System](#).

10.112.3.6 `ISystem& operator= ( const ISystem & ) [protected]`

10.112.3.7 `virtual void RegisterInterfaceEvent ( Event & evtToRegister, bool updateInterface = true ) [pure virtual]`

Implemented in [System](#).

10.112.3.8 `virtual void RegisterLoggingEvent ( LoggingEvent & handler ) [pure virtual]`

Implemented in [System](#).

10.112.3.9 `virtual void ReleaseInstance ( ) [pure virtual]`

Implemented in [System](#).

10.112.3.10 `virtual void SendActionCommand ( unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime = 0, unsigned int * pResultSize = 0, ActionCommandResult results[] = NULL ) [pure virtual]`

Implemented in [System](#).

10.112.3.11 `virtual void SetLoggingEventPriorityLevel ( SpinnakerLogLevel level ) [pure virtual]`

Implemented in [System](#).

10.112.3.12 `virtual void UnregisterAllLoggingEvent ( ) [pure virtual]`

Implemented in [System](#).

10.112.3.13 `virtual void UnregisterInterfaceEvent ( Event & evtToUnregister )` [pure virtual]

Implemented in [System](#).

10.112.3.14 `virtual void UnregisterLoggingEvent ( LoggingEvent & handler )` [pure virtual]

Implemented in [System](#).

10.112.3.15 `virtual bool UpdateCameras ( bool updateInterfaces = true )` [pure virtual]

Implemented in [System](#).

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

- `include/Interface/ISystem.h`

## 10.113 JPEGOption Struct Reference

Options for saving JPEG image.

### Public Member Functions

- [JPEGOption](#) ()

### Public Attributes

- bool [progressive](#)  
*Whether to save as a progressive JPEG file.*
- unsigned int [quality](#)  
*JPEG image quality in range (0-100).*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

#### 10.113.1 Detailed Description

Options for saving JPEG image.

#### 10.113.2 Constructor & Destructor Documentation

10.113.2.1 `JPEGOption ( )` [inline]

#### 10.113.3 Member Data Documentation

10.113.3.1 `bool progressive`

Whether to save as a progressive JPEG file.



### 10.113.3.2 unsigned int quality

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

### 10.113.3.3 unsigned int reserved[16]

Reserved for future use.

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

- include/[SpinnakerDefs.h](#)

## 10.114 JPG2Option Struct Reference

Options for saving JPEG2000 image.

### Public Member Functions

- [JPG2Option](#) ()

### Public Attributes

- unsigned int [quality](#)  
*JPEG saving quality in range (1-512).*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 10.114.1 Detailed Description

Options for saving JPEG2000 image.

### 10.114.2 Constructor & Destructor Documentation

#### 10.114.2.1 [JPG2Option](#) ( ) `[inline]`

### 10.114.3 Member Data Documentation

#### 10.114.3.1 unsigned int quality

JPEG saving quality in range (1-512).

#### 10.114.3.2 unsigned int reserved[16]

Reserved for future use.

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

- include/[SpinnakerDefs.h](#)

### 10.115 LibraryVersion Struct Reference

Provides easier access to the current version of [Spinnaker](#).

#### Public Attributes

- unsigned int [major](#)  
*Major version of the library.*
- unsigned int [minor](#)  
*Minor version of the library.*
- unsigned int [type](#)  
*Version type of the library.*
- unsigned int [build](#)  
*Build number of the library.*

#### 10.115.1 Detailed Description

Provides easier access to the current version of [Spinnaker](#).

#### 10.115.2 Member Data Documentation

##### 10.115.2.1 unsigned int build

Build number of the library.

##### 10.115.2.2 unsigned int major

Major version of the library.

##### 10.115.2.3 unsigned int minor

Minor version of the library.

## 10.115.2.4 unsigned int type

Version type of the library.

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

- [include/SpinnakerDefs.h](#)

## 10.116 LockableObject< Object >::Lock Class Reference

A scopelevel [Lock](#) class.

### Public Member Functions

- [Lock](#) (const [LockableObject](#)< Object > &obj)
- [~Lock](#) ()

### 10.116.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >::Lock
```

A scopelevel [Lock](#) class.

Automatically acquires the lock when created and releases it when destroyed.

### 10.116.2 Constructor & Destructor Documentation

10.116.2.1 [Lock](#) ( const [LockableObject](#)< Object > & *obj* ) [inline]

10.116.2.2 [~Lock](#) ( ) [inline]

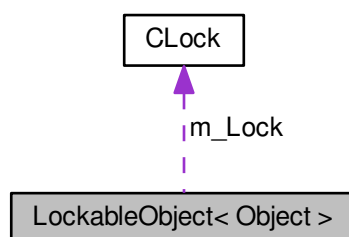
The documentation for this class was generated from the following file:

- [include/SpinGenApi/GCSynch.h](#)

## 10.117 LockableObject< Object > Class Template Reference

Instance-Lock for an object.

Collaboration diagram for LockableObject< Object >:



### Classes

- class [Lock](#)  
A scopelevel [Lock](#) class.

### Public Member Functions

- [Lock GetLock](#) () const  
Get a new lock.

### Public Attributes

- [CLock m\\_Lock](#)

### Friends

- class [Lock](#)

### 10.117.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >
```

Instance-Lock for an object.

## 10.117.2 Member Function Documentation

### 10.117.2.1 Lock GetLock ( ) const [inline]

Get a new lock.

## 10.117.3 Friends And Related Function Documentation

### 10.117.3.1 friend class Lock [friend]

## 10.117.4 Member Data Documentation

### 10.117.4.1 CLock m\_Lock [mutable]

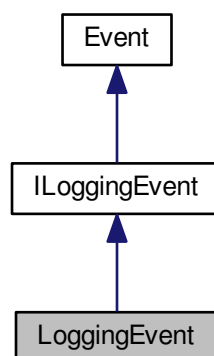
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

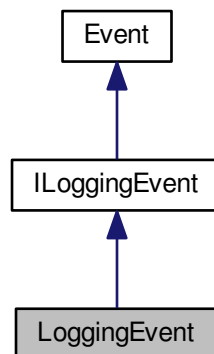
## 10.118 LoggingEvent Class Reference

An event handler for capturing the device logging event.

Inheritance diagram for LoggingEvent:



Collaboration diagram for LoggingEvent:



## Public Member Functions

- [LoggingEvent](#) ()  
*Default constructor.*
- [~LoggingEvent](#) ()  
*Virtual destructor.*
- virtual void [OnLogEvent](#) ([LoggingEventDataPtr](#) eventPtr)=0  
*The callback for the log event.*

## Protected Member Functions

- [LoggingEvent](#) & [operator=](#) (const [LoggingEvent](#) &)  
*Assignment operator.*

## Additional Inherited Members

### 10.118.1 Detailed Description

An event handler for capturing the device logging event.

### 10.118.2 Constructor & Destructor Documentation

#### 10.118.2.1 [LoggingEvent](#) ( )

Default constructor.

10.118.2.2 `~LoggingEvent ( )`

Virtual destructor.

## 10.118.3 Member Function Documentation

10.118.3.1 `virtual void OnLogEvent ( LoggingEventDataPtr eventPtr )` [pure virtual]

The callback for the log event.

## Parameters

|                       |                           |
|-----------------------|---------------------------|
| <code>eventPtr</code> | The logging event pointer |
|-----------------------|---------------------------|

Implements [ILoggingEvent](#).

10.118.3.2 `LoggingEvent& operator= ( const LoggingEvent & )` [protected]

Assignment operator.

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

- include/[LoggingEvent.h](#)

## 10.119 LoggingEventData Class Reference

The [LoggingEventData](#) object.

## Public Member Functions

- [~LoggingEventData \( \)](#)  
*Default Destructor.*
- `const char *` [GetCategoryName \( \)](#)  
*Gets the logging event category name.*
- `const char *` [GetLogMessage \( \)](#)  
*Gets the logging event message.*
- `const char *` [GetNDC \( \)](#)  
*Gets the logging event's Nested Diagnostic Context (NDC).*
- `const int` [GetPriority \( \)](#)  
*Gets the logging event priority.*
- `const char *` [GetThreadName \( \)](#)  
*Gets the logging event thread name.*
- `const char *` [GetTimestamp \( \)](#)  
*Gets the logging event time stamp.*
- `const char *` [GetPriorityName \( \)](#)  
*Gets the logging event priority name.*

## Protected Member Functions

- [LoggingEventData](#) (void \*data)  
*Default Constructor.*

## Friends

- class [SystemImpl](#)

### 10.119.1 Detailed Description

The [LoggingEventData](#) object.

### 10.119.2 Constructor & Destructor Documentation

#### 10.119.2.1 ~LoggingEventData ( )

Default Destructor.

#### 10.119.2.2 LoggingEventData ( void \* *data* ) [protected]

Default Constructor.

### 10.119.3 Member Function Documentation

#### 10.119.3.1 const char\* GetCategoryName ( )

Gets the logging event category name.

##### Returns

The category name

#### 10.119.3.2 const char\* GetLogMessage ( )

Gets the logging event message.

##### Returns

The log message



### 10.119.3.3 `const char* GetNDC ( )`

Gets the logging event's Nested Diagnostic Context (NDC).

#### Returns

The log event's NDC

### 10.119.3.4 `const int GetPriority ( )`

Gets the logging event priority.

#### Returns

The log priority

### 10.119.3.5 `const char* GetPriorityName ( )`

Gets the logging event priority name.

#### Returns

The priority name of the log

### 10.119.3.6 `const char* GetThreadName ( )`

Gets the logging event thread name.

#### Returns

The thread name

### 10.119.3.7 `const char* GetTimestamp ( )`

Gets the logging event time stamp.

#### Returns

The time stamp of the log

## 10.119.4 Friends And Related Function Documentation

### 10.119.4.1 `friend class SystemImpl [friend]`

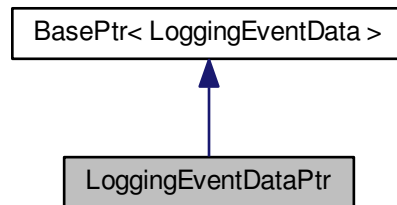
The documentation for this class was generated from the following file:

- [include/LoggingEventData.h](#)

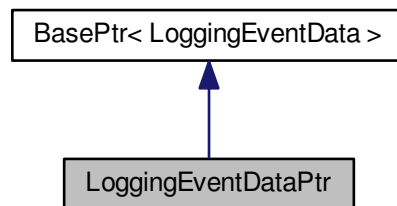
## 10.120 LoggingEventDataPtr Class Reference

A reference tracked pointer to the [LoggingEvent](#) object.

Inheritance diagram for LoggingEventDataPtr:



Collaboration diagram for LoggingEventDataPtr:



### Public Member Functions

- [LoggingEventDataPtr](#) () throw ()  
*Default Constructor.*
- [LoggingEventDataPtr](#) (const int) throw ()  
*Default Constructor.*
- virtual [LoggingEventDataPtr](#) & [operator=](#) (const int nMustBeNull)  
*Copy Constructor.*
- virtual [~LoggingEventDataPtr](#) (void)  
*Virtual Destructor.*

### Additional Inherited Members

#### 10.120.1 Detailed Description

A reference tracked pointer to the [LoggingEvent](#) object.

## 10.120.2 Constructor & Destructor Documentation

### 10.120.2.1 LoggingEventDataPtr ( ) throw [inline]

Default Constructor.

### 10.120.2.2 LoggingEventDataPtr ( const int ) throw [inline]

Default Constructor.

### 10.120.2.3 virtual ~LoggingEventDataPtr ( void ) [inline],[virtual]

Virtual Destructor.

## 10.120.3 Member Function Documentation

### 10.120.3.1 virtual LoggingEventDataPtr& operator= ( const int *nMustBeNull* ) [inline],[virtual]

Copy Constructor.

Reimplemented from [BasePtr< LoggingEventData >](#).

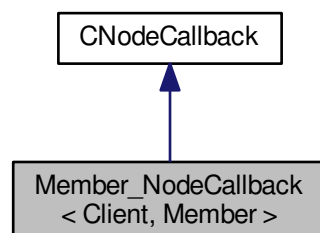
The documentation for this class was generated from the following file:

- [include/LoggingEventDataPtr.h](#)

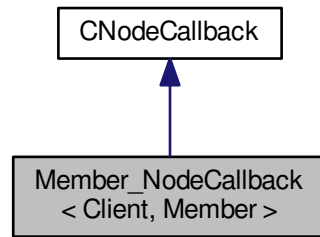
## 10.121 Member\_NodeCallback< Client, Member > Class Template Reference

Container for a member function pointer.

Inheritance diagram for Member\_NodeCallback< Client, Member >:



Collaboration diagram for Member\_NodeCallback< Client, Member >:



## Public Types

- typedef void(Client::\* [PMEMBERFUNC](#)) (INode \*)  
*Member function type.*

## Public Member Functions

- [Member\\_NodeCallback](#) (INode \*pNode, Client &client, Member member, [ECallbackType](#) CallbackType)  
*Constructor.*
- virtual void [operator\(\)](#) ([ECallbackType](#) CallbackType) const  
*execute operation*
- virtual void [Destroy](#) ()  
*destroys the object*

## Additional Inherited Members

### 10.121.1 Detailed Description

```
template<class Client, class Member>
class Spinnaker::GenApi::Member_NodeCallback< Client, Member >
```

Container for a member function pointer.

### 10.121.2 Member Typedef Documentation

#### 10.121.2.1 typedef void(Client::\* PMEMBERFUNC) (INode \*)

Member function type.

### 10.121.3 Constructor & Destructor Documentation

10.121.3.1 **Member\_NodeCallback** ( *Inode \* pNode*, *Client & client*, *Member member*, *ECallbackType CallbackType* )  
[inline]

Constructor.

### 10.121.4 Member Function Documentation

10.121.4.1 **virtual void Destroy** ( ) [inline],[virtual]

destroys the object

Implements [CNodeCallback](#).

10.121.4.2 **virtual void operator()** ( *ECallbackType CallbackType* ) const [inline],[virtual]

execute operation

Implements [CNodeCallback](#).

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

- include/SpinGenApi/[NodeCallback.h](#)

## 10.122 MJPGOption Struct Reference

Options for saving MJPG files.

### Public Member Functions

- [MJPGOption](#) ()

### Public Attributes

- float [frameRate](#)  
*Frame rate of the stream.*
- unsigned int [quality](#)  
*Image quality (1-100)*
- unsigned int [reserved](#) [256]

### 10.122.1 Detailed Description

Options for saving MJPG files.

## 10.122.2 Constructor & Destructor Documentation

### 10.122.2.1 MJPGOption( ) [inline]

## 10.122.3 Member Data Documentation

### 10.122.3.1 float frameRate

Frame rate of the stream.

### 10.122.3.2 unsigned int quality

Image quality (1-100)

### 10.122.3.3 unsigned int reserved[256]

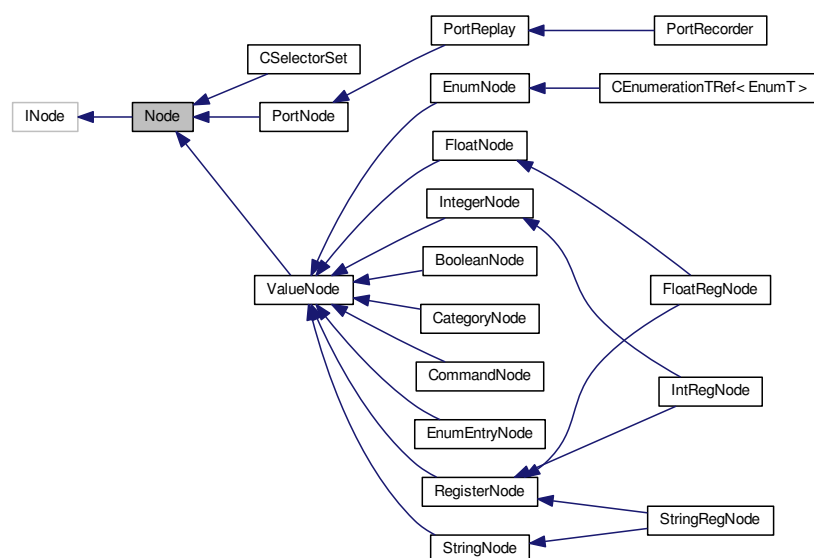
The documentation for this struct was generated from the following file:

- include/[SpinVideoDefs.h](#)

## 10.123 Node Class Reference

class common to all nodes

Inheritance diagram for Node:



Collaboration diagram for Node:



## Public Member Functions

- [Node](#) ()  
*Constructor.*
- [Node](#) (std::shared\_ptr< Node::NodeImpl > pNodeHandle)  
*Constructor.*
- [~Node](#) ()  
*Destructor.*
- virtual [GenICam::gcstring GetName](#) (bool FullQualified=false) const  
*Get node name.*
- virtual [GenApi::ENamespace GetNameSpace](#) () const  
*Get name space.*
- virtual [EVisibility GetVisibility](#) () const  
*Get the recommended visibility of the node.*
- virtual void [InvalidateNode](#) ()  
*Indicates that the node's value may have changed.*
- virtual bool [IsCacheable](#) () const  
*Is the node value cacheable.*
- virtual [EYesNo IsAccessModeCacheable](#) () const  
*True if the AccessMode can be cached.*
- virtual [ECachingMode GetCachingMode](#) () const  
*Get Caching Mode.*
- virtual int64\_t [GetPollingTime](#) () const  
*recommended polling time (for not cacheable nodes)*
- virtual [GenICam::gcstring GetToolTip](#) () const  
*Get a short description of the node.*
- virtual [GenICam::gcstring GetDescription](#) () const  
*Get a long description of the node.*
- virtual [GenICam::gcstring GetDisplayName](#) () const  
*Get a name string for display.*
- virtual [GenICam::gcstring GetDeviceName](#) () const  
*Get a name of the device.*
- virtual void [GetChildren](#) (GenApi::NodeList\_t &Children, [ELinkType](#) LinkType=ctReadingChildren) const  
*Get all nodes this node directly depends on.*
- virtual void [GetParents](#) (GenApi::NodeList\_t &Parents) const

- Gets all nodes this node is directly depending on.*

  - virtual [CallbackHandleType RegisterCallback](#) ([CNodeCallback](#) \*pCallback)

*Register change callback Takes ownership of the [CNodeCallback](#) object.*
- virtual bool [DeregisterCallback](#) ([CallbackHandleType](#) hCallback)

*De register change callback Destroys [CNodeCallback](#) object.*
- virtual [INodeMap](#) \* [GetNodeMap](#) () const

*Retrieves the central node map.*
- virtual [GenICam::gcstring](#) [GetEventID](#) () const

*Get the EventId of the node.*
- virtual bool [IsStreamable](#) () const

*True if the node is streamable.*
- virtual void [GetPropertyNames](#) ([GenICam::gcstring\\_vector](#) &PropertyNames) const

*Returns a list of the names all properties set during initialization.*
- virtual bool [GetProperty](#) (const [GenICam::gcstring](#) &PropertyName, [GenICam::gcstring](#) &ValueStr, [GenICam::gcstring](#) &AttributeStr)

*Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.*
- virtual void [ImposeAccessMode](#) ([EAccessMode](#) ImposedAccessMode)

*Imposes an access mode to the natural access mode of the node.*
- virtual void [ImposeVisibility](#) ([EVisibility](#) ImposedVisibility)

*Imposes a visibility to the natural visibility of the node.*
- virtual [INode](#) \* [GetAlias](#) () const

*Retrieves the a node which describes the same feature in a different way.*
- virtual [INode](#) \* [GetCastAlias](#) () const

*Retrieves the a node which describes the same feature so that it can be casted.*
- virtual [GenICam::gcstring](#) [GetDocuURL](#) () const

*Gets a URL pointing to the documentation of that feature.*
- virtual bool [IsDeprecated](#) () const

*True if the node should not be used any more.*
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const

*Get the type of the main interface of a node.*
- virtual bool [IsFeature](#) () const

*True if the node can be reached via category nodes from a category node named "Root".*
- void [SetNodeHandle](#) (std::shared\_ptr< [Node::NodeImpl](#) > pNodeHandle)

*Set [Node](#) handle.*
- std::shared\_ptr< [Node::NodeImpl](#) > [GetNodeHandle](#) () const

*Get [Node](#) handle.*
- virtual [EAccessMode](#) [GetAccessMode](#) () const

*Base interface overrides.*
- virtual bool [IsSelector](#) () const

*Selector interface overrides.*
- virtual void [GetSelectedFeatures](#) ([FeatureList\\_t](#) &) const

*retrieve the group of selected features*
- virtual void [GetSelectingFeatures](#) ([FeatureList\\_t](#) &) const

*retrieve the group of features selecting this node*
- virtual void [SetReference](#) ([INode](#) \*pBase)

*Reference interface overrides [\ingroup Spinnaker\\_GenApi\\_PublicImpl](#).*
- virtual void [SetReference](#) ([ISelector](#) \*pBase)
- void [SetNodeMap](#) ([INodeMap](#) \*pNodeMap)
- virtual bool [operator==](#) (int nullPtr) const
- virtual bool [operator!=](#) (int nullPtr) const



## Protected Attributes

- `std::shared_ptr< Node::NodeImpl > m_pNodeData`
- `std::list< CallbackHandleType_t * > m_Callbacks`  
*List of callbacks.*
- `INodeMap * m_pNodeMap`

### 10.123.1 Detailed Description

class common to all nodes

### 10.123.2 Constructor & Destructor Documentation

#### 10.123.2.1 `Node ( )`

Constructor.

#### 10.123.2.2 `Node ( std::shared_ptr< Node::NodeImpl > pNodeHandle )`

Constructor.

#### 10.123.2.3 `~Node ( )`

Destructor.

### 10.123.3 Member Function Documentation

#### 10.123.3.1 `virtual bool DeregisterCallback ( CallbackHandleType hCallback ) [virtual]`

De register change callback Destroys [CNodeCallback](#) object.

##### Returns

true if the callback handle was valid

#### 10.123.3.2 `virtual EAccessMode GetAccessMode ( ) const [virtual]`

Base interface overrides.

Get the access mode of the node

Reimplemented in [PortRecorder](#).

10.123.3.3 `virtual INode* GetAlias ( ) const [virtual]`

Retrieves the a node which describes the same feature in a different way.

10.123.3.4 `virtual ECachingMode GetCachingMode ( ) const [virtual]`

Get Caching Mode.

10.123.3.5 `virtual INode* GetCastAlias ( ) const [virtual]`

Retrieves the a node which describes the same feature so that it can be casted.

10.123.3.6 `virtual void GetChildren ( GenApi::NodeList_t & Children, ELinkType LinkType = ctReadingChildren ) const [virtual]`

Get all nodes this node directly depends on.

#### Parameters

|     |                 |                        |
|-----|-----------------|------------------------|
| out | <i>Children</i> | List of children nodes |
|     | <i>LinkType</i> | The link type          |

10.123.3.7 `virtual GenICam::gcstring GetDescription ( ) const [virtual]`

Get a long description of the node.

10.123.3.8 `virtual GenICam::gcstring GetDeviceName ( ) const [virtual]`

Get a name of the device.

10.123.3.9 `virtual GenICam::gcstring GetDisplayName ( ) const [virtual]`

Get a name string for display.

10.123.3.10 `virtual GenICam::gcstring GetDocuURL ( ) const [virtual]`

Gets a URL pointing to the documentation of that feature.

10.123.3.11 `virtual GenICam::gcstring GetEventID ( ) const [virtual]`

Get the EventId of the node.

10.123.3.12 `virtual GenICam::gcstring GetName ( bool FullQualified = false ) const` [virtual]

Get node name.

10.123.3.13 `virtual GenApi::ENamespace GetNamespace ( ) const` [virtual]

Get name space.

10.123.3.14 `std::shared_ptr<Node::NodeImpl> GetNodeHandle ( ) const`

Get [Node](#) handle.

10.123.3.15 `virtual INodeMap* GetNodeMap ( ) const` [virtual]

Retrieves the central node map.

10.123.3.16 `virtual void GetParents ( GenApi::NodeList_t & Parents ) const` [virtual]

Gets all nodes this node is directly depending on.

Parameters

|     |                |                      |
|-----|----------------|----------------------|
| out | <i>Parents</i> | List of parent nodes |
|-----|----------------|----------------------|

10.123.3.17 `virtual int64_t GetPollingTime ( ) const` [virtual]

recommended polling time (for not cacheable nodes)

10.123.3.18 `virtual EInterfaceType GetPrincipalInterfaceType ( ) const` [virtual]

Get the type of the main interface of a node.

10.123.3.19 `virtual bool GetProperty ( const GenICam::gcstring & PropertyName, GenICam::gcstring & ValueStr,  
GenICam::gcstring & AttributeStr )` [virtual]

Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.

10.123.3.20 `virtual void GetPropertyNames ( GenICam::gcstring_vector & PropertyNames ) const` [virtual]

Returns a list of the names all properties set during initialization.

10.123.3.21 `virtual void GetSelectedFeatures ( FeatureList_t & ) const [virtual]`

retrieve the group of selected features

10.123.3.22 `virtual void GetSelectingFeatures ( FeatureList_t & ) const [virtual]`

retrieve the group of features selecting this node

10.123.3.23 `virtual GenlCam::gcstring GetToolTip ( ) const [virtual]`

Get a short description of the node.

10.123.3.24 `virtual EVisibility GetVisibility ( ) const [virtual]`

Get the recommended visibility of the node.

10.123.3.25 `virtual void ImposeAccessMode ( EAccessMode ImposedAccessMode ) [virtual]`

Imposes an access mode to the natural access mode of the node.

10.123.3.26 `virtual void ImposeVisibility ( EVisibility ImposedVisibility ) [virtual]`

Imposes a visibility to the natural visibility of the node.

10.123.3.27 `virtual void InvalidateNode ( ) [virtual]`

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

10.123.3.28 `virtual EYesNo IsAccessModeCacheable ( ) const [virtual]`

True if the AccessMode can be cached.

10.123.3.29 `virtual bool IsCachable ( ) const [virtual]`

Is the node value cacheable.

10.123.3.30 `virtual bool IsDeprecated ( ) const [virtual]`

True if the node should not be used any more.

10.123.3.31 `virtual bool IsFeature ( ) const [virtual]`

True if the node can be reached via category nodes from a category node named "Root".

10.123.3.32 `virtual bool IsSelector ( ) const [virtual]`

Selector interface overrides.

true if this feature selects a group of features

10.123.3.33 `virtual bool IsStreamable ( ) const [virtual]`

True if the node is streamable.

10.123.3.34 `virtual bool operator!= ( int nullPtr ) const [virtual]`

10.123.3.35 `virtual bool operator== ( int nullPtr ) const [virtual]`

10.123.3.36 `virtual CallbackHandleType RegisterCallback ( CNodeCallback * pCallback ) [virtual]`

Register change callback Takes ownership of the [CNodeCallback](#) object.

10.123.3.37 `void SetNodeHandle ( std::shared_ptr< Node::NodeImpl > pNodeHandle )`

Set [Node](#) handle.

10.123.3.38 `void SetNodeMap ( INodeMap * pNodeMap )`

10.123.3.39 `virtual void SetReference ( INode * pBase ) [virtual]`

Reference interface overrides `lingroup Spinnaker_GenApi_PublicImpl`.

Reimplemented in [FloatNode](#), [PortNode](#), [IntegerNode](#), [EnumNode](#), [CEnumerationTRef< EnumT >](#), [StringNode](#), [ValueNode](#), [RegisterNode](#), [BooleanNode](#), [CommandNode](#), [EnumEntryNode](#), [CategoryNode](#), [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

10.123.3.40 `virtual void SetReference ( ISelector * pBase ) [virtual]`

## 10.123.4 Member Data Documentation

10.123.4.1 `std::list<CallbackHandleType_t*> m_Callbacks [protected]`

List of callbacks.

10.123.4.2 `std::shared_ptr<Node::NodeImpl> m_pNodeData` [protected]

10.123.4.3 `INodeMap* m_pNodeMap` [protected]

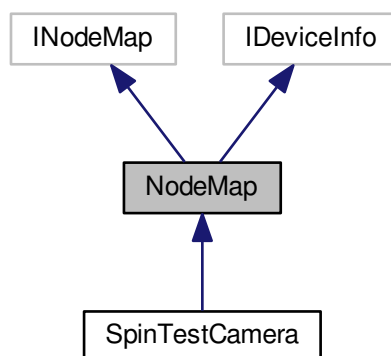
The documentation for this class was generated from the following file:

- `include/SpinGenApi/Node.h`

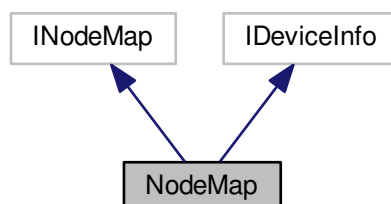
## 10.124 NodeMap Class Reference

Smart pointer template for NodeMaps with create function.

Inheritance diagram for NodeMap:



Collaboration diagram for NodeMap:



## Public Member Functions

- [NodeMap](#) ([GenICam::gcstring](#) DeviceName="Device")  
*Constructor.*
- virtual [~NodeMap](#) ()  
*Destructor.*
- void [Destroy](#) ()  
*Destroys the node map.*
- void [LoadXMLFromFile](#) ([GenICam::gcstring](#) FileName)  
*Creates the object from a XML file with given file name.*
- void [LoadXMLFromZIPFile](#) ([GenICam::gcstring](#) ZipFileName)  
*Creates the object from a ZIP'd XML file with given file name.*
- void [LoadXMLFromZIPData](#) (const void \*zipData, size\_t zipSize)  
*Creates the object from a ZIP'd XML file given in a string.*
- void [LoadXMLFromFileInject](#) ([GenICam::gcstring](#) TargetFileName, [GenICam::gcstring](#) InjectFileName)  
*Creates the object from a XML target and an inject file with given file name.*
- void [LoadXMLFromString](#) (const [GenICam::gcstring](#) &XMLData)  
*Creates the object from XML data given in a string.*
- void [LoadXMLFromStringInject](#) (const [GenICam::gcstring](#) &TargetXMLDataconst, const [GenICam::gcstring](#) &InjectXMLData)  
*Creates the object from XML data given in a string with injection.*
- virtual void [GetSupportedSchemaVersions](#) ([GenICam::gcstring\\_vector](#) &SchemaVersions)  
*Gets a list of supported schema versions.*
- virtual [GenICam::gcstring](#) [GetDeviceName](#) ()  
*Get device name.*
- virtual void [Poll](#) (int64\_t ElapsedTime)  
*Fires nodes which have a polling time.*
- virtual void [GetNodes](#) ([NodeList\\_t](#) &Nodes) const  
*Retrieves all nodes in the node map.*
- virtual [INode](#) \* [GetNode](#) (const [GenICam::gcstring](#) &key) const  
*Retrieves the node from the central map by name.*
- virtual void [InvalidateNodes](#) () const  
*Invalidates all nodes.*
- virtual bool [Connect](#) ([IPort](#) \*pPort, const [GenICam::gcstring](#) &PortName) const  
*Connects a port to a port node with given name.*
- virtual bool [Connect](#) ([IPort](#) \*pPort) const  
*Connects a port to the standard port "Device".*
- virtual [CLock](#) & [GetLock](#) () const  
*Returns the lock which guards the node map.*
- virtual uint64\_t [GetNumNodes](#) () const  
*Get the number of nodes in the map.*
- void \* [GetNodeMapHandle](#) () const
- virtual [GenICam::gcstring](#) [GetModelName](#) ()  
*Get the model name.*
- virtual [GenICam::gcstring](#) [GetVendorName](#) ()  
*Get the vendor name.*
- virtual [GenICam::gcstring](#) [GetToolTip](#) ()  
*Get tool tip.*
- virtual [GenICam::gcstring](#) [GetStandardNameSpace](#) ()  
*Get the standard name space.*
- virtual void [GetGenApiVersion](#) ([GenICam::Version\\_t](#) &Version, uint16\_t &Build)

Get the version of the DLL's *GenApi* implementation.

- virtual void `GetSchemaVersion (GenICam::Version_t &Version)`

Get the schema version number.

- virtual void `GetDeviceVersion (GenICam::Version_t &Version)`

Get the version of the device description file.

- virtual `GenICam::gcstring GetProductGuid ()`

Get the GUID describing the product.

- virtual `GenICam::gcstring GetVersionGuid ()`

Get the GUID describing the product version.

## Static Public Member Functions

- static bool `ClearXMLCache ()`

Clears the cache of the camera description files.

## Public Attributes

- `INodeMap * _Ptr`

Pointer to the *NodeMap*.

### 10.124.1 Detailed Description

Smart pointer template for NodeMaps with create function.

#### Parameters

|                      |                                                                           |
|----------------------|---------------------------------------------------------------------------|
| <i>TCameraParams</i> | The camera specific parameter class (auto generated from camera xml file) |
|----------------------|---------------------------------------------------------------------------|

### 10.124.2 Constructor & Destructor Documentation

#### 10.124.2.1 `NodeMap ( GenICam::gcstring DeviceName = "Device" )`

Constructor.

#### 10.124.2.2 `virtual ~NodeMap ( ) [virtual]`

Destructor.

### 10.124.3 Member Function Documentation

#### 10.124.3.1 `static bool ClearXMLCache ( ) [static]`

Clears the cache of the camera description files.



10.124.3.2 `virtual bool Connect ( IPort * pPort, const GenICam::gcstring & PortName ) const` [virtual]

Connects a port to a port node with given name.

10.124.3.3 `virtual bool Connect ( IPort * pPort ) const` [virtual]

Connects a port to the standard port "Device".

10.124.3.4 `void Destroy ( )`

Destroys the node map.

10.124.3.5 `virtual GenICam::gcstring GetDeviceName ( )` [virtual]

Get device name.

10.124.3.6 `virtual void GetDeviceVersion ( GenICam::Version_t & Version )` [virtual]

Get the version of the device description file.

10.124.3.7 `virtual void GetGenApiVersion ( GenICam::Version_t & Version, uint16_t & Build )` [virtual]

Get the version of the DLL's [GenApi](#) implementation.

10.124.3.8 `virtual CLock& GetLock ( ) const` [virtual]

Returns the lock which guards the node map.

10.124.3.9 `virtual GenICam::gcstring GetModelName ( )` [virtual]

Get the model name.

10.124.3.10 `virtual INode* GetNode ( const GenICam::gcstring & key ) const` [virtual]

Retrieves the node from the central map by name.

10.124.3.11 `void* GetNodeMapHandle ( ) const`

10.124.3.12 `virtual void GetNodes ( NodeList_t & Nodes ) const` [virtual]

Retrieves all nodes in the node map.

10.124.3.13 `virtual uint64_t GetNumNodes ( ) const [virtual]`

Get the number of nodes in the map.

10.124.3.14 `virtual GenICam::gcstring GetProductGuid ( ) [virtual]`

Get the GUID describing the product.

10.124.3.15 `virtual void GetSchemaVersion ( GenICam::Version_t & Version ) [virtual]`

Get the schema version number.

10.124.3.16 `virtual GenICam::gcstring GetStandardNameSpace ( ) [virtual]`

Get the standard name space.

10.124.3.17 `virtual void GetSupportedSchemaVersions ( GenICam::gcstring_vector & SchemaVersions ) [virtual]`

Gets a list of supported schema versions.

! Loads an XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromFile(const [GenICam::gcstring&](#) XMLFileName, const [GenICam::gcstring&](#) StyleSheetFileName, const [GenICam::gcstring&](#) OutputFileName, const uint32\_t XMLValidation = xvDefault);

! Loads a Zipped XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromZIPFile(const [GenICam::gcstring&](#) ZIPFileName, const [GenICam::gcstring&](#) StyleSheetFileName, const [GenICam::gcstring&](#) OutputFileName, const uint32\_t XMLValidation = xvDefault);

! Injects an XML file into a target file virtual void MergeXMLFiles( const [GenICam::gcstring&](#) TargetFileName, \*< Name of the target XML file to process const [GenICam::gcstring&](#) InjectedFileName, \*< Name of the Injected XML file to process const [GenICam::gcstring&](#) OutputFileName \*< Name of the output file );

! Extract independent subtree virtual void ExtractIndependentSubtree( const [GenICam::gcstring&](#) XMLData, \*< The XML data the subtree is extracted from. const [GenICam::gcstring&](#) InjectXMLData, \*< Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed. const [GenICam::gcstring&](#) SubTreeRootNodeName,\*< The name of the node that represents the root of the subtree that shall be extracted. [GenICam::gcstring&](#) ExtractedSubtree \*< The returned extracted subtree as string. );

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

10.124.3.18 `virtual GenICam::gcstring GetToolTip ( ) [virtual]`

Get tool tip.

10.124.3.19 **virtual GenICam::gcstring GetVendorName ( )** [virtual]

Get the vendor name.

10.124.3.20 **virtual GenICam::gcstring GetVersionGuid ( )** [virtual]

Get the GUID describing the product version.

10.124.3.21 **virtual void InvalidateNodes ( ) const** [virtual]

Invalidates all nodes.

10.124.3.22 **void LoadXMLFromFile ( GenICam::gcstring *FileName* )**

Creates the object from a XML file with given file name.

! Creates the object from the default DLL ! note Can only be used if the class TCameraParams was auto generated from a specific camera xml file void LoadDLL(void);

! Creates the object from a DLL whose name is deduced from vendor and model name void LoadDLL(GenICam↔::gcstring VendorName, GenICam::gcstring ModelName);

! Creates the object from a DLL with given file name void LoadDLL(GenICam::gcstring FileName);

10.124.3.23 **void LoadXMLFromFileInject ( GenICam::gcstring *TargetFileName*, GenICam::gcstring *InjectFileName* )**

Creates the object from a XML target and an inject file with given file name.

10.124.3.24 **void LoadXMLFromString ( const GenICam::gcstring & *XMLData* )**

Creates the object from XML data given in a string.

10.124.3.25 **void LoadXMLFromStringInject ( const GenICam::gcstring & *TargetXMLData*const, const GenICam::gcstring & *InjectXMLData* )**

Creates the object from XML data given in a string with injection.

10.124.3.26 **void LoadXMLFromZIPData ( const void \* *zipData*, size\_t *zipSize* )**

Creates the object from a ZIP'd XML file given in a string.

10.124.3.27 **void LoadXMLFromZIPFile ( GenICam::gcstring *ZipFileName* )**

Creates the object from a ZIP'd XML file with given file name.

10.124.3.28 `virtual void Poll ( int64_t ElapsedTime ) [virtual]`

Fires nodes which have a polling time.

#### 10.124.4 Member Data Documentation

10.124.4.1 `INodeMap* _Ptr`

Pointer to the [NodeMap](#).

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

- `include/SpinGenApi/NodeMap.h`

### 10.125 CNodeMapFactory::NodeStatistics\_t Struct Reference

#### Public Attributes

- `uint32_t` [NumNodes](#)
- `uint32_t` [NumProperties](#)
- `uint32_t` [NumLinks](#)
- `uint32_t` [NumStrings](#)

#### 10.125.1 Member Data Documentation

10.125.1.1 `uint32_t` [NumLinks](#)

10.125.1.2 `uint32_t` [NumNodes](#)

10.125.1.3 `uint32_t` [NumProperties](#)

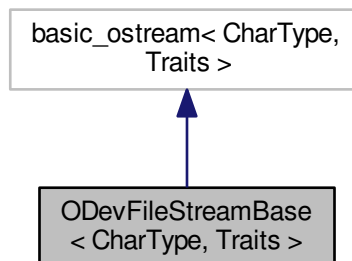
10.125.1.4 `uint32_t` [NumStrings](#)

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

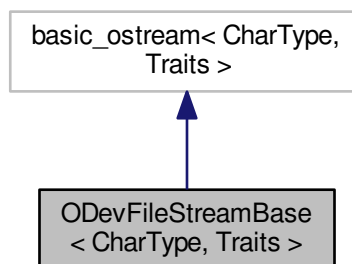
- `include/SpinGenApi/NodeMapFactory.h`

## 10.126 ODevFileStreamBase< CharType, Traits > Class Template Reference

Inheritance diagram for ODevFileStreamBase< CharType, Traits >:



Collaboration diagram for ODevFileStreamBase< CharType, Traits >:



### Public Types

- typedef [ODevFileStreamBuf< CharType, Traits >](#) [filebuf\\_type](#)
- typedef std::basic\_ios< CharType, Traits > [ios\\_type](#)
- typedef std::basic\_ostream< CharType, Traits > [ostream\\_type](#)

### Public Member Functions

- [filebuf\\_type](#) \* [rdbuf](#) () const
- bool [is\\_open](#) () const
- void [open](#) (INodeMap \*pInterface, const char \*pFileName, std::ios\_base::openmode mode=std::ios\_base::out|std::ios\_base::trunc)  
*Open file on device in write mode.*
- void [close](#) ()  
*Close the file on device.*

### 10.126.1 Member Typedef Documentation

10.126.1.1 `typedef ODevFileStreamBuf<CharType, Traits> filebuf_type`

10.126.1.2 `typedef std::basic_ios<CharType, Traits> ios_type`

10.126.1.3 `typedef std::basic_ostream<CharType, Traits> ostream_type`

### 10.126.2 Member Function Documentation

10.126.2.1 `void close ( ) [inline]`

Close the file on device.

10.126.2.2 `bool is_open ( ) const [inline]`

10.126.2.3 `void open ( INodeMap * pInterface, const char * pFileName, std::ios_base::openmode mode = std::ios_base::out | std::ios_base::trunc ) [inline]`

Open file on device in write mode.

#### Parameters

|                   |                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------|
| <i>pInterface</i> | <a href="#">NodeMap</a> of the device to which the <a href="#">FileProtocolAdapter</a> is attached |
| <i>pFileName</i>  | Name of the file to open                                                                           |
| <i>mode</i>       | open mode                                                                                          |

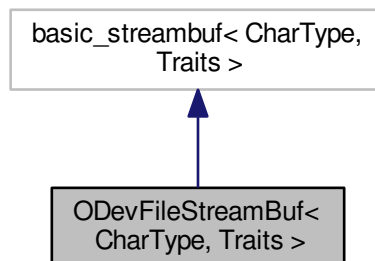
10.126.2.4 `filebuf_type* rdbuf ( ) const [inline]`

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

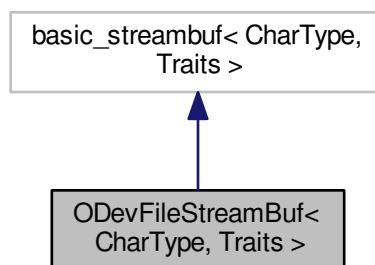
- `include/SpinGenApi/Filestream.h`

## 10.127 ODevFileStreamBuf< CharType, Traits > Class Template Reference

Inheritance diagram for ODevFileStreamBuf< CharType, Traits >:



Collaboration diagram for ODevFileStreamBuf< CharType, Traits >:



### Public Member Functions

- [ODevFileStreamBuf](#) ()
- [~ODevFileStreamBuf](#) ()
- [filebuf\\_type](#) \* [open](#) ([Spinnaker::GenApi::INodeMap](#) \*pInterface, const char \*pFileName, std::ios\_base::openmode mode)
- bool [is\\_open](#) () const
- [filebuf\\_type](#) \* [close](#) ()

### Protected Member Functions

- std::streamsize [xsputn](#) (const char\_type \*s, std::streamsize n)
- int\_type [overflow](#) (int\_type c=traits\_type::eof())
- int [sync](#) ()

### 10.127.1 Constructor & Destructor Documentation

10.127.1.1 `ODevFileStreamBuf ( )` `[inline]`

10.127.1.2 `~ODevFileStreamBuf ( )` `[inline]`

### 10.127.2 Member Function Documentation

10.127.2.1 `filebuf_type* close ( )` `[inline]`

10.127.2.2 `bool is_open ( ) const` `[inline]`

10.127.2.3 `filebuf_type* open ( Spinnaker::GenApi::INodeMap * pInterface, const char * pFileName, std::ios_base::openmode mode )` `[inline]`

10.127.2.4 `int_type overflow ( int_type c = traits_type::eof() )` `[inline]`, `[protected]`

10.127.2.5 `int sync ( )` `[inline]`, `[protected]`

10.127.2.6 `std::streamsize xsputn ( const char_type * s, std::streamsize n )` `[inline]`, `[protected]`

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

- `include/SpinGenApi/Filestream.h`

## 10.128 PGMOption Struct Reference

Options for saving PGM images.

### Public Member Functions

- `PGMOption ( )`

### Public Attributes

- `bool binaryFile`  
*Whether to save the PPM as a binary file.*
- `unsigned int reserved [16]`  
*Reserved for future use.*

### 10.128.1 Detailed Description

Options for saving PGM images.



## 10.128.2 Constructor & Destructor Documentation

### 10.128.2.1 PGMOption ( ) `[inline]`

## 10.128.3 Member Data Documentation

### 10.128.3.1 bool binaryFile

Whether to save the PPM as a binary file.

### 10.128.3.2 unsigned int reserved[16]

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

## 10.129 PNGOption Struct Reference

Options for saving PNG images.

### Public Member Functions

- [PNGOption \( \)](#)

### Public Attributes

- bool [interlaced](#)  
*Whether to save the PNG as interlaced.*
- unsigned int [compressionLevel](#)  
*Compression level (0-9).*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 10.129.1 Detailed Description

Options for saving PNG images.

## 10.129.2 Constructor & Destructor Documentation

### 10.129.2.1 PNGOption ( ) [inline]

## 10.129.3 Member Data Documentation

### 10.129.3.1 unsigned int compressionLevel

Compression level (0-9).

0 is no compression, 9 is best compression.

### 10.129.3.2 bool interlaced

Whether to save the PNG as interlaced.

### 10.129.3.3 unsigned int reserved[16]

Reserved for future use.

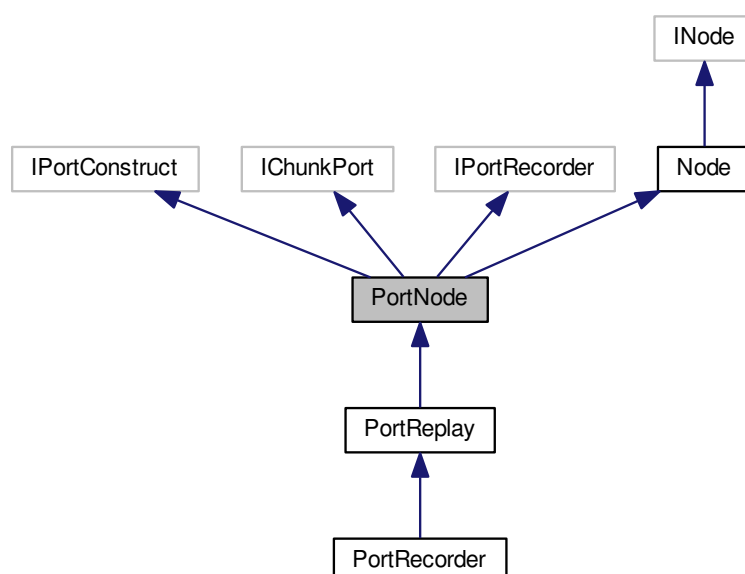
The documentation for this struct was generated from the following file:

- include/[SpinnakerDefs.h](#)

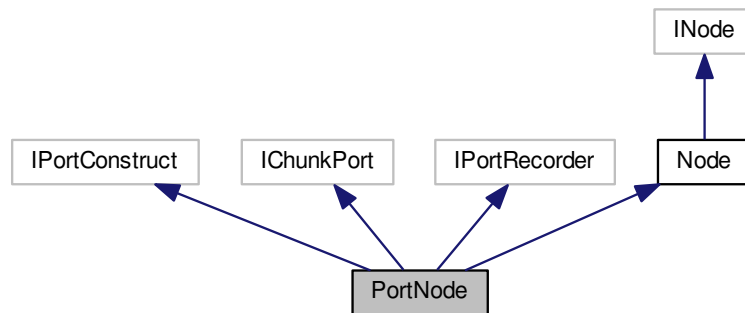
## 10.130 PortNode Class Reference

[Interface](#) for value properties.

Inheritance diagram for PortNode:



Collaboration diagram for PortNode:



## Public Member Functions

- [PortNode](#) ()  
*Constructor.*
- [PortNode](#) (std::shared\_ptr< Node::NodeImpl > pValue)  
*constructor with [GenICam](#) IValue*
- [~PortNode](#) ()  
*Destructor.*
- virtual void [Read](#) (void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Reads a chunk of bytes from the port.*
- virtual void [Write](#) (const void \*pBuffer, int64\_t [Address](#), int64\_t [Length](#))  
*Writes a chunk of bytes to the port.*
- void [SetPortImpl](#) (IPort \*pPort)  
*Sets pointer the real port implementation; this function may called only once.*
- virtual [EYesNo](#) [GetSwapEndianness](#) ()  
*Determines if the port adapter must perform an endianness swap.*
- virtual [Spinnaker::GenICam::gcstring](#) [GetChunkID](#) () const  
*Get the Id of the chunk the port should be attached to.*
- virtual [EYesNo](#) [CacheChunkData](#) () const  
*Indicates if the chunk a adapter must hold a cached version of the chunk data.*
- virtual void [StartRecording](#) (IPortWriteList \*pPortRecorder)  
*Starts logging all WriteRegister commands to a list.*
- virtual void [StopRecording](#) ()  
*Stops recording.*
- virtual void [Replay](#) (IPortWriteList \*pPortRecorder, bool [Invalidate](#)=true)  
*Sends the commands to the camera.*
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Value*
- virtual void [SetReference](#) (IPort \*pBase)  
*overload SetReference for Value*
- virtual void [SetReference](#) (IChunkPort \*pBase)  
*overload SetReference for Value*
- std::shared\_ptr< Node::NodeImpl > [GetPortHandle](#) ()

## Additional Inherited Members

### 10.130.1 Detailed Description

[Interface](#) for value properties.

### 10.130.2 Constructor & Destructor Documentation

#### 10.130.2.1 PortNode ( )

Constructor.

#### 10.130.2.2 PortNode ( std::shared\_ptr< Node::NodeImpl > pValue )

constructor with [GenICam](#) IValue

#### 10.130.2.3 ~PortNode ( )

Destructor.

### 10.130.3 Member Function Documentation

#### 10.130.3.1 virtual EYesNo CacheChunkData ( ) const [virtual]

Indicates if the chunk a adapter must hold a cached version of the chunk data.

#### 10.130.3.2 virtual Spinnaker::GenICam::gcstring GetChunkID ( ) const [virtual]

Get the Id of the chunk the port should be attached to.

#### 10.130.3.3 std::shared\_ptr<Node::NodeImpl> GetPortHandle ( ) [inline]

#### 10.130.3.4 virtual EYesNo GetSwapEndianness ( ) [virtual]

Determines if the port adapter must perform an endianness swap.

#### 10.130.3.5 virtual void Read ( void \* pBuffer, int64\_t Address, int64\_t Length ) [virtual]

Reads a chunk of bytes from the port.

10.130.3.6 `virtual void Replay ( IPortWriteList * pPortRecorder, bool Invalidate = true ) [virtual]`

Sends the commands to the camera.

The default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented in [PortReplay](#).

10.130.3.7 `void SetPortImpl ( IPort * pPort )`

Sets pointer the real port implementation; this function may called only once.

10.130.3.8 `virtual void SetReference ( INode * pBase ) [virtual]`

overload SetReference for Value

Reimplemented from [Node](#).

10.130.3.9 `virtual void SetReference ( IPort * pBase ) [virtual]`

overload SetReference for Value

Reimplemented in [PortRecorder](#), and [PortReplay](#).

10.130.3.10 `virtual void SetReference ( IChunkPort * pBase ) [virtual]`

overload SetReference for Value

10.130.3.11 `virtual void StartRecording ( IPortWriteList * pPortRecorder ) [virtual]`

Starts logging all WriteRegister commands to a list.

Reimplemented in [PortRecorder](#).

10.130.3.12 `virtual void StopRecording ( ) [virtual]`

Stops recording.

Reimplemented in [PortRecorder](#).

10.130.3.13 `virtual void Write ( const void * pBuffer, int64_t Address, int64_t Length )` [virtual]

Writes a chunk of bytes to the port.

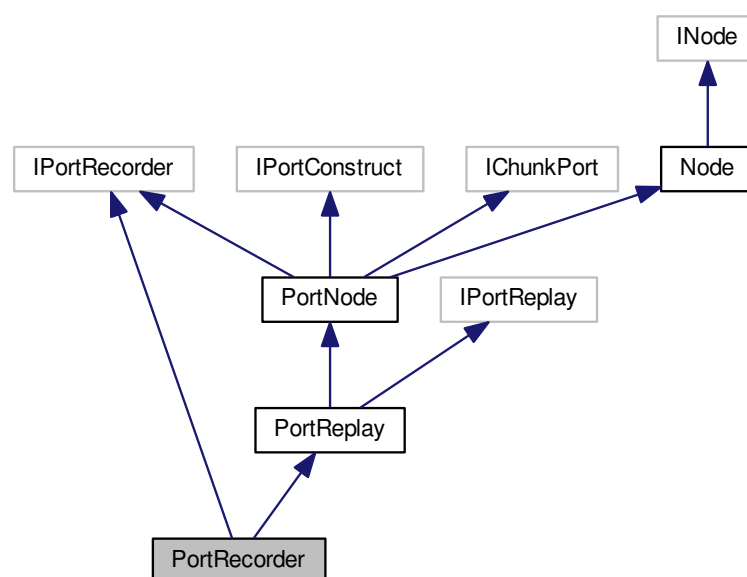
The documentation for this class was generated from the following file:

- `include/SpinGenApi/PortNode.h`

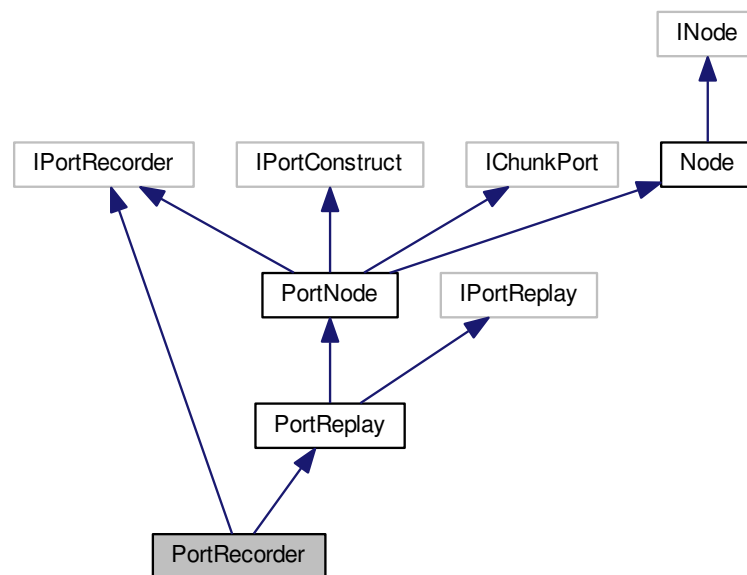
## 10.131 PortRecorder Class Reference

[Interface](#) for recording write commands on a port.

Inheritance diagram for PortRecorder:



Collaboration diagram for PortRecorder:



## Public Member Functions

- [PortRecorder](#) ()
- virtual [~PortRecorder](#) ()
- virtual void [StartRecording](#) ([IPortWriteList](#) \*pPortRecorder)  
*starts logging all WriteRegister commands to a list*
- virtual void [StopRecording](#) ()  
*stops recording*
- virtual [EAccessMode](#) [GetAccessMode](#) () const  
*Get the access mode of the node.*
- virtual void [SetReference](#) ([IPort](#) \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.131.1 Detailed Description

[Interface](#) for recording write commands on a port.

## 10.131.2 Constructor & Destructor Documentation

### 10.131.2.1 PortRecorder ( )

### 10.131.2.2 virtual ~PortRecorder ( ) [virtual]

## 10.131.3 Member Function Documentation

### 10.131.3.1 virtual EAccessMode GetAccessMode ( ) const [virtual]

Get the access mode of the node.

Reimplemented from [Node](#).

### 10.131.3.2 virtual void SetReference ( IPort \* *pBase* ) [virtual]

overload SetReference for Value

Reimplemented from [PortReplay](#).

### 10.131.3.3 virtual void StartRecording ( IPortWriteList \* *pPortRecorder* ) [virtual]

starts logging all WriteRegister commands to a list

Reimplemented from [PortNode](#).

### 10.131.3.4 virtual void StopRecording ( ) [virtual]

stops recording

Reimplemented from [PortNode](#).

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

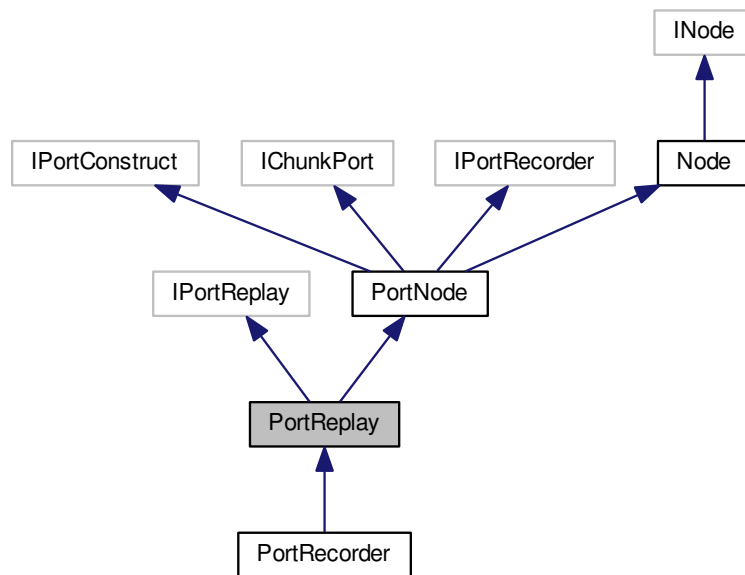
- include/SpinGenApi/[PortRecorder.h](#)



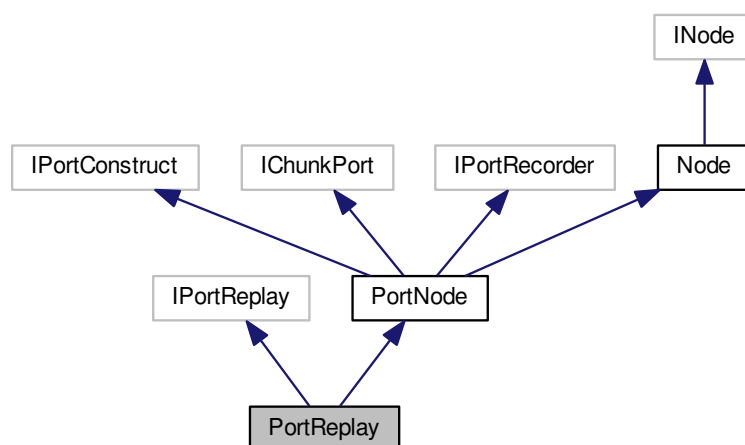
## 10.132 PortReplay Class Reference

[Interface](#) for replaying write commands on a port.

Inheritance diagram for PortReplay:



Collaboration diagram for PortReplay:



## Public Member Functions

- [PortReplay](#) ()
- virtual [~PortReplay](#) ()
- virtual void [Replay](#) ([IPortWriteList](#) \*pPortRecorder, bool [Invalidate](#)=true)  
*sends the commands to the camera.*
- virtual void [SetReference](#) ([IPort](#) \*pBase)  
*overload SetReference for Value*
- void \* [GetPortReplayHandle](#) ()

## Additional Inherited Members

### 10.132.1 Detailed Description

[Interface](#) for replaying write commands on a port.

### 10.132.2 Constructor & Destructor Documentation

#### 10.132.2.1 [PortReplay](#) ( )

#### 10.132.2.2 virtual [~PortReplay](#) ( ) [virtual]

### 10.132.3 Member Function Documentation

#### 10.132.3.1 void\* [GetPortReplayHandle](#) ( )

#### 10.132.3.2 virtual void [Replay](#) ( [IPortWriteList](#) \* *pPortRecorder*, bool *Invalidate* =true ) [virtual]

sends the commands to the camera.

the default implementation just walks the list and issues each command using the [WriteRegister](#) method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented from [PortNode](#).

#### 10.132.3.3 virtual void [SetReference](#) ( [IPort](#) \* *pBase* ) [virtual]

overload [SetReference](#) for Value

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

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

- include/SpinGenApi/[PortReplay.h](#)

## 10.133 PPMOption Struct Reference

Options for saving PPM images.

### Public Member Functions

- [PPMOption](#) ()

### Public Attributes

- bool [binaryFile](#)  
*Whether to save the PPM as a binary file.*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

#### 10.133.1 Detailed Description

Options for saving PPM images.

#### 10.133.2 Constructor & Destructor Documentation

##### 10.133.2.1 [PPMOption](#) ( ) [[inline](#)]

#### 10.133.3 Member Data Documentation

##### 10.133.3.1 bool [binaryFile](#)

Whether to save the PPM as a binary file.

##### 10.133.3.2 unsigned int [reserved](#)[16]

Reserved for future use.

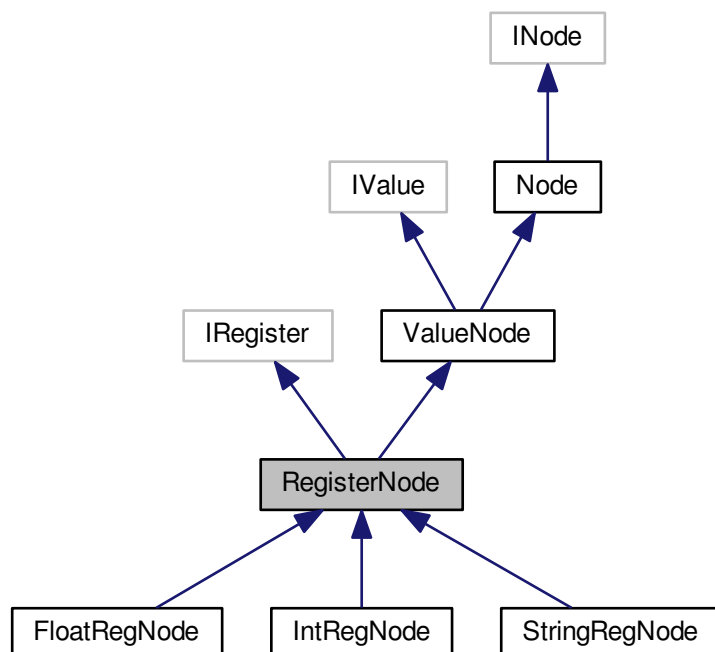
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

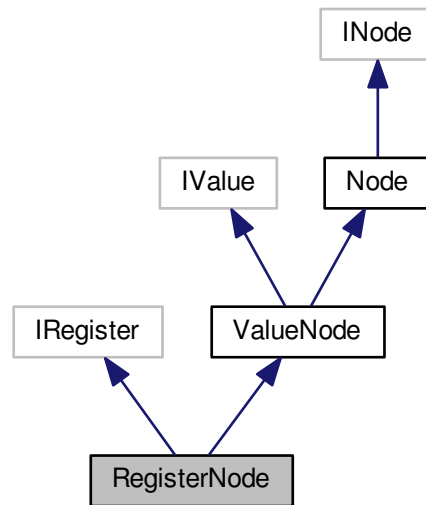
## 10.134 RegisterNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for RegisterNode:



Collaboration diagram for RegisterNode:



## Public Member Functions

- [RegisterNode](#) ()
- [RegisterNode](#) (std::shared\_ptr< Node::NodeImpl > pRegister)
- virtual [~RegisterNode](#) ()
- virtual void [Set](#) (const uint8\_t \*pBuffer, int64\_t [Length](#), bool [Verify](#)=true)  
*Set the register's contents.*
- virtual void [Get](#) (uint8\_t \*pBuffer, int64\_t [Length](#), bool [Verify](#)=false, bool IgnoreCache=false)  
*Fills a buffer with the register's contents.*
- virtual int64\_t [GetLength](#) ()  
*Retrieves the Length of the register [Bytes].*
- virtual int64\_t [GetAddress](#) ()  
*Retrieves the Address of the register.*
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Register*

## Additional Inherited Members

### 10.134.1 Detailed Description

[Interface](#) for string properties.

## 10.134.2 Constructor & Destructor Documentation

10.134.2.1 **RegisterNode** ( )

10.134.2.2 **RegisterNode** ( `std::shared_ptr< Node::NodeImpl > pRegister` )

10.134.2.3 **virtual ~RegisterNode** ( ) [virtual]

## 10.134.3 Member Function Documentation

10.134.3.1 **virtual void Get** ( `uint8_t* pBuffer`, `int64_t Length`, `bool Verify = false`, `bool IgnoreCache = false` ) [virtual]

Fills a buffer with the register's contents.

### Parameters

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| <i>pBuffer</i>     | The buffer receiving the data to read                                          |
| <i>Length</i>      | The number of bytes to retrieve                                                |
| <i>Verify</i>      | Enables Range verification (default = false). The AccessMode is always checked |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false)                |

### Returns

The value read

10.134.3.2 **virtual int64\_t GetAddress** ( ) [virtual]

Retrieves the Address of the register.

10.134.3.3 **virtual int64\_t GetLength** ( ) [virtual]

Retrieves the Length of the register [Bytes].

10.134.3.4 **virtual void Set** ( `const uint8_t* pBuffer`, `int64_t Length`, `bool Verify = true` ) [virtual]

Set the register's contents.

### Parameters

|                |                                                            |
|----------------|------------------------------------------------------------|
| <i>pBuffer</i> | The buffer containing the data to set                      |
| <i>Length</i>  | The number of bytes in pBuffer                             |
| <i>Verify</i>  | Enables AccessMode and Range verification (default = true) |

10.134.3.5 `virtual void SetReference ( INode * pBase ) [virtual]`

overload SetReference for Register

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

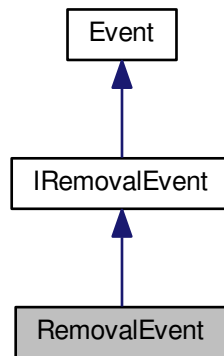
The documentation for this class was generated from the following file:

- `include/SpinGenApi/`[RegisterNode.h](#)

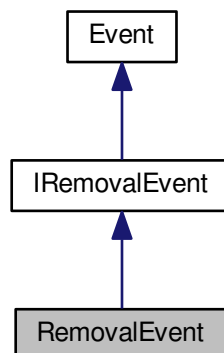
## 10.135 RemovalEvent Class Reference

An event handler for capturing the device removal event.

Inheritance diagram for RemovalEvent:



Collaboration diagram for RemovalEvent:



## Public Member Functions

- [RemovalEvent](#) ()  
*Default Constructor.*
- virtual [~RemovalEvent](#) ()  
*Virtual Destructor.*
- virtual void [OnDeviceRemoval](#) (uint64\_t serialNumber)=0  
*Device removal event callback.*

## Protected Member Functions

- [RemovalEvent](#) & [operator=](#) (const [RemovalEvent](#) &)  
*Assignment operator.*

## Additional Inherited Members

### 10.135.1 Detailed Description

An event handler for capturing the device removal event.

### 10.135.2 Constructor & Destructor Documentation

#### 10.135.2.1 [RemovalEvent](#) ( )

Default Constructor.

#### 10.135.2.2 virtual [~RemovalEvent](#) ( ) [virtual]

Virtual Destructor.

### 10.135.3 Member Function Documentation

#### 10.135.3.1 virtual void [OnDeviceRemoval](#) ( uint64\_t *serialNumber* ) [pure virtual]

Device removal event callback.

#### Parameters

|                     |                                         |
|---------------------|-----------------------------------------|
| <i>serialNumber</i> | The serial number of the device removed |
|---------------------|-----------------------------------------|

Implements [IRemovalEvent](#).



10.135.3.2 **RemovalEvent& operator=** ( const **RemovalEvent** & ) [protected]

Assignment operator.

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

- include/[RemovalEvent.h](#)

## 10.136 SingleChunkData\_t Struct Reference

### Public Attributes

- uint64\_t [ChunkID](#)
- ptrdiff\_t [ChunkOffset](#)
- size\_t [ChunkLength](#)

### 10.136.1 Member Data Documentation

10.136.1.1 uint64\_t [ChunkID](#)

10.136.1.2 size\_t [ChunkLength](#)

10.136.1.3 ptrdiff\_t [ChunkOffset](#)

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

- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

## 10.137 SingleChunkDataStr\_t Struct Reference

### Public Attributes

- GenlCam::gcstring [ChunkID](#)
- ptrdiff\_t [ChunkOffset](#)
- size\_t [ChunkLength](#)

### 10.137.1 Member Data Documentation

10.137.1.1 GenlCam::gcstring [ChunkID](#)

10.137.1.2 size\_t [ChunkLength](#)

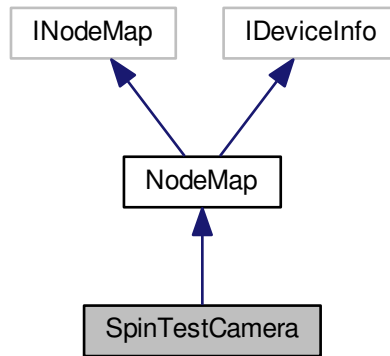
10.137.1.3 ptrdiff\_t [ChunkOffset](#)

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

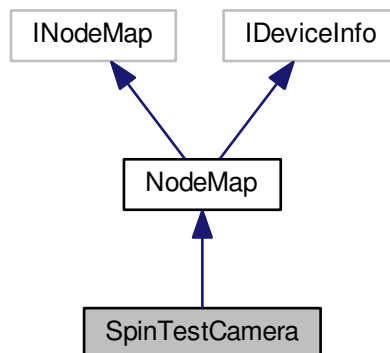
- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

## 10.138 SpinTestCamera Class Reference

Inheritance diagram for SpinTestCamera:



Collaboration diagram for SpinTestCamera:



### Additional Inherited Members

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

- include/SpinGenApi/[SpinTestCamera.h](#)

## 10.139 SpinVideo Class Reference

Provides the functionality for the user to record images to an AVI/MP4 file.

## Public Member Functions

- [SpinVideo](#) ()  
*Default constructor.*
- virtual [~SpinVideo](#) ()  
*Default destructor.*
- virtual void [Open](#) (const char \*pFileName, [AVIOption](#) &pOption)  
*Open an video file in preparation for writing Images to disk.*
- virtual void [Open](#) (const char \*pFileName, [MJPGOption](#) &pOption)  
*Open an MJPEG video file in preparation for writing Images to disk.*
- virtual void [Open](#) (const char \*pFileName, [H264Option](#) &pOption)  
*Open an H264 MP4 video file in preparation for writing Images to disk.*
- virtual void [Append](#) ([ImagePtr](#) plmage)  
*Append an image to the video file.*
- virtual void [Close](#) ()  
*Close the video file.*
- virtual void [SetMaximumFileSize](#) (unsigned int size)  
*Set the maximum file size (in megabytes) of a AVI/MP4 file.*

### 10.139.1 Detailed Description

Provides the functionality for the user to record images to an AVI/MP4 file.

### 10.139.2 Constructor & Destructor Documentation

#### 10.139.2.1 [SpinVideo](#) ( )

Default constructor.

#### 10.139.2.2 virtual [~SpinVideo](#) ( ) [virtual]

Default destructor.

### 10.139.3 Member Function Documentation

#### 10.139.3.1 virtual void [Append](#) ( [ImagePtr](#) plmage ) [virtual]

Append an image to the video file.

Parameters

|                        |                      |
|------------------------|----------------------|
| <a href="#">plmage</a> | The image to append. |
|------------------------|----------------------|

### 10.139.3.2 virtual void Close ( ) [virtual]

Close the video file.

See also

[Open\(\)](#)

### 10.139.3.3 virtual void Open ( const char \* *pFileName*, AVIOption & *pOption* ) [virtual]

Open an video file in preparation for writing Images to disk.

The size of video files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

|                  |                                     |
|------------------|-------------------------------------|
| <i>pFileName</i> | The filename of the video file.     |
| <i>pOption</i>   | Options to apply to the video file. |

See also

[Close\(\)](#)

### 10.139.3.4 virtual void Open ( const char \* *pFileName*, MJPGOption & *pOption* ) [virtual]

Open an MJPEG video file in preparation for writing Images to disk.

The size of video files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

|                  |                                           |
|------------------|-------------------------------------------|
| <i>pFileName</i> | The filename of the video file.           |
| <i>pOption</i>   | MJPEG options to apply to the video file. |

See also

[Close\(\)](#)  
[MJPGOption](#)

### 10.139.3.5 virtual void Open ( const char \* *pFileName*, H264Option & *pOption* ) [virtual]

Open an H264 MP4 video file in preparation for writing Images to disk.

The size of MP4 files is limited to 2GB. The filenames are automatically generated using the filename specified.

## Parameters

|                  |                                              |
|------------------|----------------------------------------------|
| <i>pFileName</i> | The filename of the MP4 video file.          |
| <i>pOption</i>   | H264 options to apply to the MP4 video file. |

## See also

[Close\(\)](#)  
[H264Option](#)

10.139.3.6 `virtual void SetMaximumFileSize ( unsigned int size )` `[virtual]`

Set the maximum file size (in megabytes) of a AVI/MP4 file.

A new video file is created automatically when file size limit is reached. Setting a maximum size of 0 indicates no limit on file size.

## Parameters

|             |                                    |
|-------------|------------------------------------|
| <i>size</i> | The maximum video file size in MB. |
|-------------|------------------------------------|

## See also

[Append\(ImagePtr pImage\)](#)

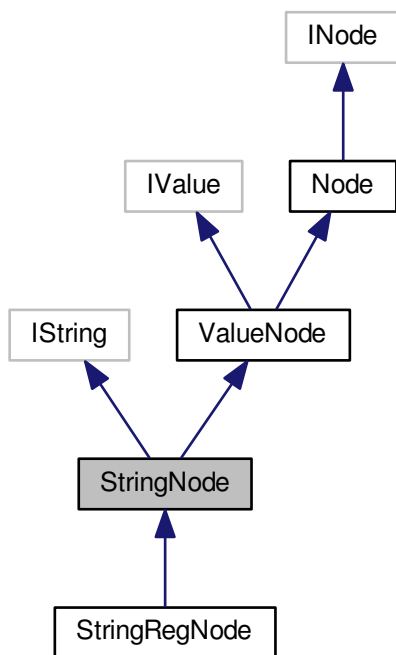
The documentation for this class was generated from the following file:

- [include/SpinVideo.h](#)

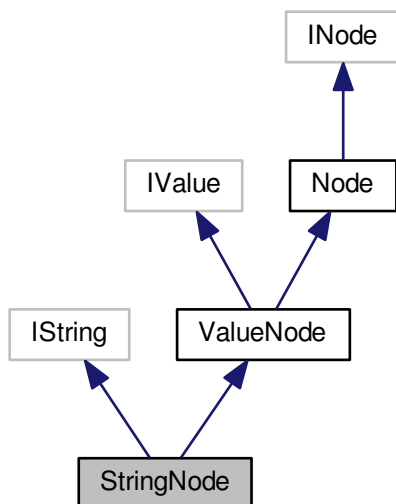
## 10.140 StringNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for StringNode:



Collaboration diagram for StringNode:



## Public Member Functions

- [StringNode](#) ()
- [StringNode](#) (std::shared\_ptr< Node::NodeImpl > pString)
- virtual [~StringNode](#) ()
- virtual void [SetValue](#) (const [GenICam::gcstring](#) &Value, bool [Verify](#)=true)  
*Set node value.*
- virtual [IString](#) & [operator=](#) (const [GenICam::gcstring](#) &Value)  
*Set node value.*
- virtual [GenICam::gcstring](#) [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)  
*Get node value.*
- virtual [GenICam::gcstring](#) [operator\(\)](#) ()  
*Get node value.*
- virtual [GenICam::gcstring](#) [operator\\*](#) ()  
*Get node value.*
- virtual int64\_t [GetMaxLength](#) ()  
*Retrieves the maximum length of the string in bytes.*
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.140.1 Detailed Description

[Interface](#) for string properties.

### 10.140.2 Constructor & Destructor Documentation

#### 10.140.2.1 [StringNode](#) ( )

#### 10.140.2.2 [StringNode](#) ( std::shared\_ptr< Node::NodeImpl > pString )

#### 10.140.2.3 virtual [~StringNode](#) ( ) [virtual]

### 10.140.3 Member Function Documentation

#### 10.140.3.1 virtual int64\_t [GetMaxLength](#) ( ) [virtual]

Retrieves the maximum length of the string in bytes.

#### 10.140.3.2 virtual [GenICam::gcstring](#) [GetValue](#) ( bool [Verify](#) = false, bool [IgnoreCache](#) = false ) [virtual]

Get node value.

## Parameters

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| <i>Verify</i>      | Enables Range verification (default = false). The AccessMode is always checked |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false)                |

## Returns

The value read

10.140.3.3 `virtual GenICam::gcstring operator()( )` [virtual]

Get node value.

10.140.3.4 `virtual GenICam::gcstring operator*( )` [virtual]

Get node value.

10.140.3.5 `virtual IString& operator=( const GenICam::gcstring & Value )` [virtual]

Set node value.

10.140.3.6 `virtual void SetReference ( INode * pBase )` [virtual]

overload SetReference for Value

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#).

10.140.3.7 `virtual void SetValue ( const GenICam::gcstring & Value, bool Verify=true )` [virtual]

Set node value.

## Parameters

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>Value</i>  | The value to set                                           |
| <i>Verify</i> | Enables AccessMode and Range verification (default = true) |

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

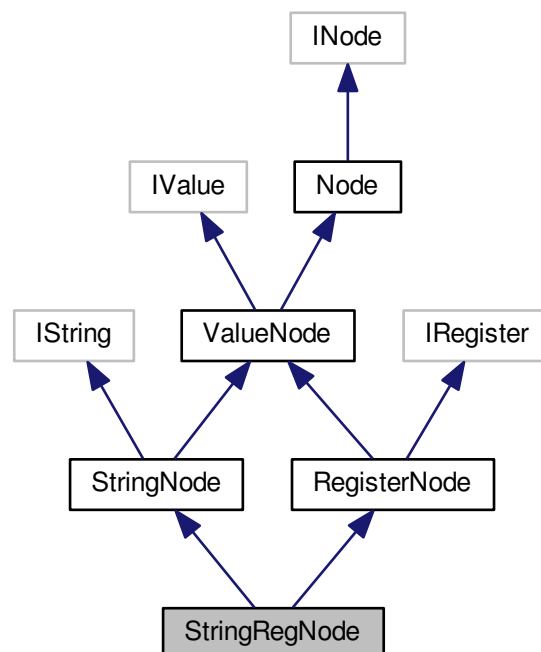
- `include/SpinGenApi/StringNode.h`



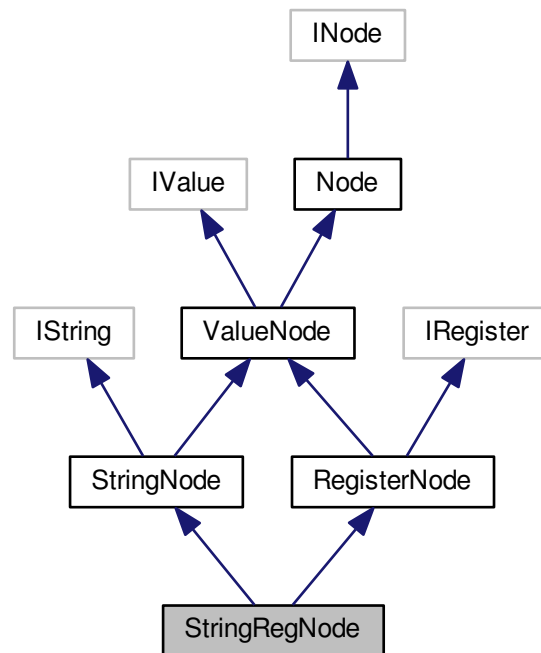
## 10.141 StringRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for StringRegNode:



Collaboration diagram for StringRegNode:



## Public Member Functions

- [StringRegNode](#) ()
- [StringRegNode](#) (std::shared\_ptr< Node::NodeImpl > pString)
- virtual [~StringRegNode](#) ()
- virtual void [SetReference](#) (INode \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.141.1 Detailed Description

[Interface](#) for string properties.

### 10.141.2 Constructor & Destructor Documentation

#### 10.141.2.1 StringRegNode ( )

#### 10.141.2.2 StringRegNode ( std::shared\_ptr< Node::NodeImpl > pString )

10.141.2.3 virtual ~StringRegNode ( ) [virtual]

### 10.141.3 Member Function Documentation

10.141.3.1 virtual void SetReference ( INode \* *pBase* ) [virtual]

overload SetReference for Value

Reimplemented from [RegisterNode](#).

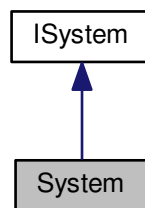
The documentation for this class was generated from the following file:

- include/SpinGenApi/[StringRegNode.h](#)

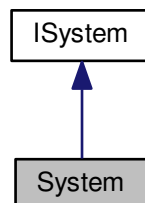
## 10.142 System Class Reference

The system object is used to retrieve the list of interfaces and cameras available.

Inheritance diagram for System:



Collaboration diagram for System:



## Public Member Functions

- virtual `~System ()`  
*Default destructor.*
- virtual void `ReleaseInstance ()`  
*This call releases the instance of the `System` Singleton for this process.*
- virtual `InterfaceList GetInterfaces (bool updateInterface=true)`  
*Returns a list of interfaces available on the system.*
- virtual `CameraList GetCameras (bool updateInterfaces=true, bool updateCameras=true)`  
*Returns a list of cameras that are available on the system.*
- virtual bool `UpdateCameras (bool updateInterfaces=true)`  
*Updates the list of cameras on the system.*
- virtual void `RegisterInterfaceEvent (Event &evtToRegister, bool updateInterface=true)`  
*Registers events for all available interfaces that are found on the system.*
- void `UnregisterInterfaceEvent (Event &evtToUnregister)`  
*Unregisters events for all available interfaces that are found on the system.*
- virtual void `RegisterLoggingEvent (LoggingEvent &handler)`  
*Registers a logging event.*
- virtual void `UnregisterAllLoggingEvent ()`  
*Unregisters all previously registered logging events.*
- virtual void `UnregisterLoggingEvent (LoggingEvent &handler)`  
*Unregisters a logging event.*
- virtual void `SetLoggingEventPriorityLevel (SpinnakerLogLevel level)`  
*Sets a threshold priority level for logging event.*
- virtual `SpinnakerLogLevel GetLoggingEventPriorityLevel ()`  
*Retrieves the current logging event priority level.*
- virtual bool `IsInUse ()`  
*Checks if the system is in use by any interface or camera objects.*
- virtual void `SendActionCommand (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL)`  
*Broadcast an Action Command to all devices on system.*
- virtual const `LibraryVersion GetLibraryVersion ()`  
*Get current library version of `Spinnaker`.*

## Static Public Member Functions

- static `SystemPtr GetInstance ()`  
*Returns a pointer to a Singleton instance of a `System` object.*

## Protected Member Functions

- `System ()`  
*Default constructor.*

### 10.142.1 Detailed Description

The system object is used to retrieve the list of interfaces and cameras available.

## 10.142.2 Constructor & Destructor Documentation

### 10.142.2.1 `virtual ~System ( ) [virtual]`

Default destructor.

### 10.142.2.2 `System ( ) [protected]`

Default constructor.

## 10.142.3 Member Function Documentation

### 10.142.3.1 `virtual CameraList GetCameras ( bool updateInterfaces = true, bool updateCameras = true ) [virtual]`

Returns a list of cameras that are available on the system.

This call returns both GigE Vision and Usb3 Vision cameras from all interfaces. The camera list object will reference count the cameras it returns. It is important that the camera list is destroyed or is cleared before calling `system->ReleaseInstance()` or else the call to `system->ReleaseInstance()` will result in an error message thrown that a reference to the camera is still held.

See also

[ReleaseInstance\(\)](#)  
[CameraList::Clear\(\)](#)

#### Parameters

|                         |                                                                                                                                       |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <i>updateInterfaces</i> | Determines whether or not <code>updateInterfaceList()</code> is called before getting cameras from available interfaces on the system |
| <i>updateCameras</i>    | Determines whether or not <code>UpdateCameras()</code> is called before getting cameras from available interfaces on the system       |

#### Returns

An [CameraList](#) object that contains a list of all cameras.

Implements [ISystem](#).

### 10.142.3.2 `static SystemPtr GetInstance ( ) [static]`

Returns a pointer to a Singleton instance of a [System](#) object.

The [System](#) object may be used to get cameras or interfaces. When an application is done using the cameras it is necessary to free the [System](#) by calling [ReleaseInstance\(\)](#).

See also

[ReleaseInstance\(\)](#)

Returns

A const ref to a system object.

10.142.3.3 `virtual InterfaceList GetInterfaces ( bool updateInterface =true ) [virtual]`

Returns a list of interfaces available on the system.

This call returns GigE and Usb2 and Usb3 interfaces.

Parameters

|                                     |                                                                                                            |
|-------------------------------------|------------------------------------------------------------------------------------------------------------|
| <code><i>updateInterface</i></code> | Determines whether or not <code>UpdateInterfaceList()</code> is called before getting available interfaces |
|-------------------------------------|------------------------------------------------------------------------------------------------------------|

Returns

An [InterfaceList](#) object that contains a list of all interfaces.

Implements [ISystem](#).

10.142.3.4 `virtual const LibraryVersion GetLibraryVersion ( ) [virtual]`

Get current library version of [Spinnaker](#).

Returns

A struct containing the current version of [Spinnaker](#) (major, minor, type, build).

Implements [ISystem](#).

10.142.3.5 `virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ( ) [virtual]`

Retrieves the current logging event priority level.

[Spinnaker](#) uses five levels of logging:

- Error - failures that are non-recoverable without user intervention.
- Warning - failures that are recoverable without user intervention.
- Notice - information about events such as camera arrival and removal, initialization and deinitialization, starting and stopping image acquisition, and feature modification.
- Info - information about recurring events that are generated regularly such as information on individual images.
- Debug - information that can be used to troubleshoot the system.

See also

[SpinnakerLogLevel](#)

Returns

Level The threshold level

Implements [ISystem](#).

10.142.3.6 `virtual bool IsInUse ( ) [virtual]`

Checks if the system is in use by any interface or camera objects.

#### Returns

Returns true if the system is in use and false otherwise.

Implements [ISystem](#).

10.142.3.7 `virtual void RegisterInterfaceEvent ( Event & evtToRegister, bool updateInterface =true ) [virtual]`

Registers events for all available interfaces that are found on the system.

#### Parameters

|                        |                                                                                                                           |
|------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <i>evtToRegister</i>   | The event to register for the available interfaces                                                                        |
| <i>updateInterface</i> | Determines whether or not UpdateInterfaceList() is called before registering event for available interfaces on the system |

Implements [ISystem](#).

10.142.3.8 `virtual void RegisterLoggingEvent ( LoggingEvent & handler ) [virtual]`

Registers a logging event.

#### Parameters

|                |                                       |
|----------------|---------------------------------------|
| <i>handler</i> | The logging event handler to register |
|----------------|---------------------------------------|

Implements [ISystem](#).

10.142.3.9 `virtual void ReleaseInstance ( ) [virtual]`

This call releases the instance of the [System](#) Singleton for this process.

After successfully releasing the [System](#) instance the pointer returned by [GetInstance\(\)](#) will be invalid. Calling ReleaseInstance while a camera reference is still held will throw an error of type SPINNAKER\_ERR\_RESOURCE\_IN\_USE.

#### See also

[Error  
GetInstance\(\)](#)

Implements [ISystem](#).

10.142.3.10 `virtual void SendActionCommand ( unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime = 0, unsigned int * pResultSize = 0, ActionCommandResult results[ ] = NULL ) [virtual]`

Broadcast an Action Command to all devices on system.

#### Parameters

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>deviceKey</i>   | The Action Command's device key                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>groupKey</i>    | The Action Command's group key                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <i>groupMask</i>   | The Action Command's group mask                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>actionTime</i>  | (Optional) Time when to assert a future action. Zero means immediate action.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>pResultSize</i> | (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results.                                                                                                                                                                                                                                                                                                                                |
| <i>results</i>     | (Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL. |

Implements [ISystem](#).

10.142.3.11 `virtual void SetLoggingEventPriorityLevel ( SpinnakerLogLevel level ) [virtual]`

Sets a threshold priority level for logging event.

Logging events below such level will not trigger callbacks.

[Spinnaker](#) uses five levels of logging:

- Error - failures that are non-recoverable without user intervention.
- Warning - failures that are recoverable without user intervention.
- Notice - information about events such as camera arrival and removal, initialization and deinitialization, starting and stopping image acquisition, and feature modification.
- Info - information about recurring events that are generated regularly such as information on individual images.
- Debug - information that can be used to troubleshoot the system.

See also

[SpinnakerLogLevel](#)

#### Parameters

|              |                     |
|--------------|---------------------|
| <i>level</i> | The threshold level |
|--------------|---------------------|

Implements [ISystem](#).



10.142.3.12 `virtual void UnregisterAllLoggingEvent ( ) [virtual]`

Unregisters all previously registered logging events.

Implements [ISystem](#).

10.142.3.13 `void UnregisterInterfaceEvent ( Event & evtToUnregister ) [virtual]`

Unregisters events for all available interfaces that are found on the system.

Parameters

|                        |                                                       |
|------------------------|-------------------------------------------------------|
| <i>evtToUnregister</i> | The event to unregister from the available interfaces |
|------------------------|-------------------------------------------------------|

Implements [ISystem](#).

10.142.3.14 `virtual void UnregisterLoggingEvent ( LoggingEvent & handler ) [virtual]`

Unregisters a logging event.

Parameters

|                |                                         |
|----------------|-----------------------------------------|
| <i>handler</i> | The logging event handler to unregister |
|----------------|-----------------------------------------|

Implements [ISystem](#).

10.142.3.15 `virtual bool UpdateCameras ( bool updateInterfaces = true ) [virtual]`

Updates the list of cameras on the system.

Note that [System::GetCameras\(\)](#) internally calls [UpdateCameras\(\)](#) for each interface it enumerates. If the list changed between this call and the last time [UpdateCameras](#) was called then the return value will be true, otherwise it is false.

See also

[GetCameras\(\)](#)

Parameters

|                         |                                                                                                                                          |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <i>updateInterfaces</i> | Determines whether or not <a href="#">UpdateInterfaceList()</a> is called before updating cameras for available interfaces on the system |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------|

Returns

True if cameras changed on interface and false otherwise.

Implements [ISystem](#).

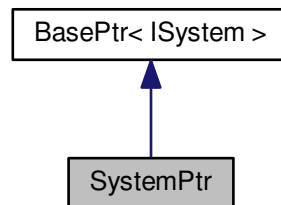
The documentation for this class was generated from the following file:

- [include/System.h](#)

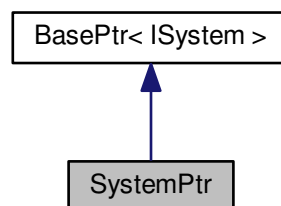
## 10.143 SystemPtr Class Reference

A reference tracked pointer to a system object.

Inheritance diagram for SystemPtr:



Collaboration diagram for SystemPtr:



### Public Member Functions

- [SystemPtr](#) ()  
*Default constructor.*
- [SystemPtr](#) (const int)  
*Copy constructor.*
- virtual [~SystemPtr](#) (void)  
*Virtual destructor.*

## Additional Inherited Members

### 10.143.1 Detailed Description

A reference tracked pointer to a system object.

### 10.143.2 Constructor & Destructor Documentation

#### 10.143.2.1 `SystemPtr ( )`

Default constructor.

#### 10.143.2.2 `SystemPtr ( const int )`

Copy constructor.

#### 10.143.2.3 `virtual ~SystemPtr ( void ) [virtual]`

Virtual destructor.

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

- `include/SystemPtr.h`

## 10.144 TIFFOption Struct Reference

Options for saving TIFF images.

### Public Types

- `enum CompressionMethod {  
    NONE = 1,  
    PACKBITS,  
    DEFLATE,  
    ADOBE_DEFLATE,  
    CCITTFAX3,  
    CCITTFAX4,  
    LZW,  
    JPEG }`

### Public Member Functions

- `TIFFOption ( )`

## Public Attributes

- [CompressionMethod compression](#)  
*Compression method to use for encoding TIFF images.*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 10.144.1 Detailed Description

Options for saving TIFF images.

### 10.144.2 Member Enumeration Documentation

#### 10.144.2.1 enum `CompressionMethod`

Enumerator

- NONE*** Save without any compression.
- PACKBITS*** Save using PACKBITS compression.
- DEFLATE*** Save using DEFLATE compression (ZLIB compression).
- ADOBE\_DEFLATE*** Save using ADOBE DEFLATE compression.
- CCITTFAX3*** Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
- CCITTFAX4*** Save using CCITT Group 4 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
- LZW*** Save using LZW compression.
- JPEG*** Save using JPEG compression. This is only valid for 8-bit greyscale and 24-bit only. Default to LZW for other bit depths.

### 10.144.3 Constructor & Destructor Documentation

#### 10.144.3.1 `TIFFOption ( )` [`inline`]

### 10.144.4 Member Data Documentation

#### 10.144.4.1 `CompressionMethod compression`

Compression method to use for encoding TIFF images.

#### 10.144.4.2 unsigned int `reserved`[16]

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

## 10.145 TransportLayerDevice Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

### Public Member Functions

- [TransportLayerDevice](#) ([GenApi::INodeMap](#) \*nodeMapTLDevice)
- [~TransportLayerDevice](#) ()

### Public Attributes

- [GenApi::IString](#) & [DeviceID](#)  
*Description: Interface-wide unique identifier of this device.*
- [GenApi::IString](#) & [DeviceSerialNumber](#)  
*Description: Serial number of the remote device.*
- [GenApi::IString](#) & [DeviceVendorName](#)  
*Description: Name of the remote device vendor.*
- [GenApi::IString](#) & [DeviceModelName](#)  
*Description: Name of the remote device model.*
- [GenApi::IEnumerationT](#)< [DeviceTypeEnum](#) > & [DeviceType](#)  
*Description: Transport layer type of the device.*
- [GenApi::IString](#) & [DeviceDisplayName](#)  
*Description: User readable name of the device.*
- [GenApi::IEnumerationT](#)< [DeviceAccessStatusEnum](#) > & [DeviceAccessStatus](#)  
*Description: Gets the access status the transport layer Producer has on the device.*
- [GenApi::IString](#) & [DeviceVersion](#)  
*Description: Version of the device.*
- [GenApi::IString](#) & [DeviceUserID](#)  
*Description: User Defined Name.*
- [GenApi::IString](#) & [DeviceDriverVersion](#)  
*Description: Version of the device driver.*
- [GenApi::IBoolean](#) & [DeviceIsUpdater](#)  
*Description: Indicates whether the device is in updater mode.*
- [GenApi::IEnumerationT](#)< [GevCCPEnum](#) > & [GevCCP](#)  
*Description: Controls the device access privilege of an application.*
- [GenApi::IEnumerationT](#)< [GUIXMLLocationEnum](#) > & [GUIXMLLocation](#)  
*Description: Sets the location to load GUI XML.*
- [GenApi::IString](#) & [GUIXMLPath](#)  
*Description: GUI XML Path.*
- [GenApi::IEnumerationT](#)< [GenICamXMLLocationEnum](#) > & [GenICamXMLLocation](#)  
*Description: Sets the location to load [GenICam](#) XML.*
- [GenApi::IString](#) & [GenICamXMLPath](#)  
*Description: [GenICam](#) XML Path.*
- [GenApi::Integer](#) & [GevDeviceIPAddress](#)  
*Description: Current IP address of the GVCP interface of the selected remote device.*
- [GenApi::Integer](#) & [GevDeviceSubnetMask](#)  
*Description: Current subnet mask of the GVCP interface of the selected remote device.*
- [GenApi::Integer](#) & [GevDeviceMACAddress](#)  
*Description: 48-bit MAC address of the GVCP interface of the selected remote device.*

- [GenApi::Integer](#) & [GevDeviceGateway](#)  
Description: Current gateway IP address of the GVCP interface of the remote device.
- [GenApi::Integer](#) & [DeviceLinkSpeed](#)  
Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.
- [GenApi::Integer](#) & [GevVersionMajor](#)  
Description: Major version of the specification.
- [GenApi::Integer](#) & [GevVersionMinor](#)  
Description: Minor version of the specification.
- [GenApi::Boolean](#) & [GevDeviceModelsBigEndian](#)  
Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).
- [GenApi::Integer](#) & [GevDeviceReadAndWriteTimeout](#)  
Description: The timeout in us for read/write operations to the camera.
- [GenApi::Integer](#) & [GevDeviceMaximumRetryCount](#)  
Description: Maximum number of times to retry a read/write operation.
- [GenApi::Integer](#) & [GevDevicePort](#)  
Description: Current IP port of the GVCP interface of the selected remote device.
- [GenApi::Command](#) & [GevDeviceDiscoverMaximumPacketSize](#)  
Description: Discovers and updates the maximum packet size that can be safely used by the device on the current interface.
- [GenApi::Integer](#) & [GevDeviceMaximumPacketSize](#)  
Description: The maximum packet size that can be safely used by the device on the current interface.
- [GenApi::Boolean](#) & [GevDeviceWrongSubnet](#)  
Description: Indicates whether the device is on the wrong subnet.
- [GenApi::Boolean](#) & [DeviceMulticastMonitorMode](#)  
Description: Controls and indicates if the device is operating in as a Multicast Monitor.
- [GenApi::EnumerationT](#) < [DeviceEndiannessMechanismEnum](#) > & [DeviceEndiannessMechanism](#)  
Description: Identifies the endianness handling mode.
- [GenApi::String](#) & [DeviceInstanceId](#)  
Description: Visibility: Invisible.
- [GenApi::EnumerationT](#) < [DeviceCurrentSpeedEnum](#) > & [DeviceCurrentSpeed](#)  
Description: The USB Speed that the device is currently operating at.
- [GenApi::Boolean](#) & [DeviceU3VProtocol](#)  
Description: Indicates whether the device is communicating in U3V Protocol.

## Protected Member Functions

- [TransportLayerDevice](#) ()

## Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameraInternal](#)

### 10.145.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

## 10.145.2 Constructor & Destructor Documentation

10.145.2.1 **TransportLayerDevice** ( **GenApi::INodeMap** \* *nodeMapTLDevice* )

10.145.2.2 **~TransportLayerDevice** ( )

10.145.2.3 **TransportLayerDevice** ( ) [protected]

## 10.145.3 Friends And Related Function Documentation

10.145.3.1 **friend class CameraBase** [friend]

10.145.3.2 **friend class CameraInternal** [friend]

10.145.3.3 **friend class ICameraBase** [friend]

## 10.145.4 Member Data Documentation

10.145.4.1 **GenApi::IEnumerationT<DeviceAccessStatusEnum> & DeviceAccessStatus**

Description: Gets the access status the transport layer Producer has on the device.

Visibility: Beginner

10.145.4.2 **GenApi::IEnumerationT<DeviceCurrentSpeedEnum> & DeviceCurrentSpeed**

Description: The USB Speed that the device is currently operating at.

Visibility: Expert

10.145.4.3 **GenApi::IString& DeviceDisplayName**

Description: User readable name of the device.

If this is not defined in the device this should be "VENDOR MODEL (ID)". Visibility: Expert

10.145.4.4 **GenApi::IString& DeviceDriverVersion**

Description: Version of the device driver.

Visibility: Expert

10.145.4.5 **GenApi::IEnumerationT<DeviceEndiannessMechanismEnum> & DeviceEndiannessMechanism**

Description: Identifies the endianness handling mode.

Visibility: Expert

**10.145.4.6 GenApi::IString& DeviceID**

Description: Interface-wide unique identifier of this device.

Visibility: Expert

**10.145.4.7 GenApi::IString& DeviceInstanceld**

Description: Visibility: Invisible.

**10.145.4.8 GenApi::IBoolean& DeviceIsUpdater**

Description: Indicates whether the device is in updater mode.

Visibility: Expert

**10.145.4.9 GenApi::IInteger& DeviceLinkSpeed**

Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.

Visibility: Expert

**10.145.4.10 GenApi::IString& DeviceModelName**

Description: Name of the remote device model.

Visibility: Beginner

**10.145.4.11 GenApi::IBoolean& DeviceMulticastMonitorMode**

Description: Controls and indicates if the device is operating in as a Multicast Monitor.

Visibility: Expert

**10.145.4.12 GenApi::IString& DeviceSerialNumber**

Description: Serial number of the remote device.

Visibility: Expert

**10.145.4.13 GenApi::IEnumerationT<DeviceTypeEnum>& DeviceType**

Description: Transport layer type of the device.

Visibility: Expert



**10.145.4.14 GenApi::IBoolean& DeviceU3VProtocol**

Description: Indicates whether the device is communicating in U3V Protocol.

Visibility: Expert

**10.145.4.15 GenApi::IString& DeviceUserID**

Description: User Defined Name.

Visibility: Expert

**10.145.4.16 GenApi::IString& DeviceVendorName**

Description: Name of the remote device vendor.

Visibility: Beginner

**10.145.4.17 GenApi::IString& DeviceVersion**

Description: Version of the device.

Visibility: Expert

**10.145.4.18 GenApi::IEnumerationT<GenICamXMLLocationEnum>& GenICamXMLLocation**

Description: Sets the location to load [GenICam](#) XML.

Visibility: Beginner

**10.145.4.19 GenApi::IString& GenICamXMLPath**

Description: [GenICam](#) XML Path.

Visibility: Beginner

**10.145.4.20 GenApi::IEnumerationT<GevCCPEnum>& GevCCP**

Description: Controls the device access privilege of an application.

Visibility: Beginner

**10.145.4.21 GenApi::ICommand& GevDeviceDiscoverMaximumPacketSize**

Description: Discovers and updates the maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

**10.145.4.22 GenApi::Integer&GevDeviceGateway**

Description: Current gateway IP address of the GVCP interface of the remote device.

Visibility: Expert

**10.145.4.23 GenApi::Integer&GevDeviceIPAddress**

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

**10.145.4.24 GenApi::Boolean&GevDeviceIsWrongSubnet**

Description: Indicates whether the device is on the wrong subnet.

Visibility: Expert

**10.145.4.25 GenApi::Integer&GevDeviceMACAddress**

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

**10.145.4.26 GenApi::Integer&GevDeviceMaximumPacketSize**

Description: The maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

**10.145.4.27 GenApi::Integer&GevDeviceMaximumRetryCount**

Description: Maximum number of times to retry a read/write operation.

Visibility: Expert

**10.145.4.28 GenApi::Boolean&GevDeviceModelsBigEndian**

Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).

Visibility: Expert

**10.145.4.29 GenApi::Integer & GevDevicePort**

Description: Current IP port of the GVCP interface of the selected remote device.

Visibility: Expert

**10.145.4.30 GenApi::Integer & GevDeviceReadAndWriteTimeout**

Description: The timeout in us for read/write operations to the camera.

Visibility: Expert

**10.145.4.31 GenApi::Integer & GevDeviceSubnetMask**

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

**10.145.4.32 GenApi::Integer & GevVersionMajor**

Description: Major version of the specification.

Visibility: Expert

**10.145.4.33 GenApi::Integer & GevVersionMinor**

Description: Minor version of the specification.

Visibility: Expert

**10.145.4.34 GenApi::EnumerationT<GUIXMLLocationEnum> & GUIXMLLocation**

Description: Sets the location to load GUI XML.

Visibility: Beginner

**10.145.4.35 GenApi::String & GUIXMLPath**

Description: GUI XML Path.

Visibility: Beginner

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

- include/[TransportLayerDevice.h](#)

## 10.146 TransportLayerInterface Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

### Public Member Functions

- [TransportLayerInterface](#) ([GenApi::INodeMap](#) \*nodeMapTLDevice)
- [~TransportLayerInterface](#) ()

### Public Attributes

- [GenApi::IString](#) & [InterfaceID](#)  
*Description: Transport layer Producer wide unique identifier of the selected interface.*
- [GenApi::IString](#) & [InterfaceDisplayName](#)  
*Description: User readable name of the selected interface.*
- [GenApi::IString](#) & [InterfaceType](#)  
*Description: Transport layer type of the interface.*
- [GenApi::Integer](#) & [GevInterfaceGateway](#)  
*Description: IP address of the selected gateway entry of this interface.*
- [GenApi::Integer](#) & [GevInterfaceMACAddress](#)  
*Description: 48-bit MAC address of this interface.*
- [GenApi::Integer](#) & [GevInterfaceIPAddress](#)  
*Description: IP address of the selected subnet of this interface.*
- [GenApi::Integer](#) & [GevInterfaceSubnetMask](#)  
*Description: Subnet mask of the selected subnet of this interface.*
- [GenApi::EnumerationT< POEStatusEnum >](#) & [POEStatus](#)  
*Description: Reports and controls the interface's power over Ethernet status.*
- [GenApi::Integer](#) & [GevActionDeviceKey](#)  
*Description: Key to authorize the action for the device.*
- [GenApi::Integer](#) & [GevActionGroupKey](#)  
*Description: Provides the key that the device will use to validate the action on reception of the action protocol message.*
- [GenApi::Integer](#) & [GevActionGroupMask](#)  
*Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.*
- [GenApi::Integer](#) & [GevActionTime](#)  
*Description: Provides the time in nanoseconds when the action is to be executed.*
- [GenApi::ICommand](#) & [ActionCommand](#)  
*Description: Issues an Action Command to attached GEV devices on interface.*
- [GenApi::IString](#) & [DeviceUnlock](#)  
*Description: Unlocks devices for internal use.*
- [GenApi::ICommand](#) & [DeviceUpdateList](#)  
*Description: Updates the internal device list.*
- [GenApi::Integer](#) & [DeviceCount](#)  
*Description: Number of compatible devices detected on current interface.*
- [GenApi::Integer](#) & [DeviceSelector](#)  
*Description: Selector for the different devices on this interface.*
- [GenApi::IString](#) & [DeviceID](#)  
*Description: [Interface](#) wide unique identifier of the selected device.*

- [GenApi::IString](#) & [DeviceVendorName](#)  
Description: Name of the device vendor.
- [GenApi::IString](#) & [DeviceModelName](#)  
Description: Name of the device model.
- [GenApi::IEnumerationT](#) < [DeviceAccessStatusEnum](#) > & [DeviceAccessStatus](#)  
Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".
- [GenApi::Integer](#) & [GevDeviceIPAddress](#)  
Description: Current IP address of the GVCP interface of the selected remote device.
- [GenApi::Integer](#) & [GevDeviceSubnetMask](#)  
Description: Current subnet mask of the GVCP interface of the selected remote device.
- [GenApi::Integer](#) & [GevDeviceMACAddress](#)  
Description: 48-bit MAC address of the GVCP interface of the selected remote device.
- [GenApi::ICommand](#) & [AutoForceIP](#)  
Description: Automatically forces any cameras on interface to an IP Address on the same subnet as the interface.
- [GenApi::Integer](#) & [IncompatibleDeviceCount](#)  
Description: Number of incompatible devices detected on current interface.
- [GenApi::Integer](#) & [IncompatibleDeviceSelector](#)  
Description: Selector for the devices that are not compatible with [Spinnaker](#) on this interface.
- [GenApi::IString](#) & [IncompatibleDeviceID](#)  
Description: [Interface](#) wide unique identifier of the selected incompatible device.
- [GenApi::IString](#) & [IncompatibleDeviceVendorName](#)  
Description: Name of the incompatible device vendor.
- [GenApi::IString](#) & [IncompatibleDeviceModelName](#)  
Description: Name of the incompatible device model.
- [GenApi::Integer](#) & [IncompatibleGevDeviceIPAddress](#)  
Description: Current IP address of the GVCP interface of the selected remote incompatible device.
- [GenApi::Integer](#) & [IncompatibleGevDeviceSubnetMask](#)  
Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.
- [GenApi::Integer](#) & [IncompatibleGevDeviceMACAddress](#)  
Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.
- [GenApi::IString](#) & [HostAdapterName](#)  
Description: User readable name of the interface's host adapter.
- [GenApi::IString](#) & [HostAdapterVendor](#)  
Description: User readable name of the host adapter's vendor.
- [GenApi::IString](#) & [HostAdapterDriverVersion](#)  
Description: Driver version of the interface's host adapter.

## Protected Member Functions

- [TransportLayerInterface](#) ()

## Friends

- class [Interface](#)
- class [IInterface](#)
- class [InterfaceInternal](#)

### 10.146.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

### 10.146.2 Constructor & Destructor Documentation

10.146.2.1 **TransportLayerInterface** ( **GenApi::INodeMap** \* *nodeMapTLDevice* )

10.146.2.2 **~TransportLayerInterface** ( )

10.146.2.3 **TransportLayerInterface** ( ) [protected]

### 10.146.3 Friends And Related Function Documentation

10.146.3.1 **friend class IInterface** [friend]

10.146.3.2 **friend class Interface** [friend]

10.146.3.3 **friend class InterfaceInternal** [friend]

### 10.146.4 Member Data Documentation

10.146.4.1 **GenApi::ICommand& ActionCommand**

Description: Issues an Action Command to attached GEV devices on interface.

Visibility: Expert

10.146.4.2 **GenApi::ICommand& AutoForceIP**

Description: Automatically forces any cameras on interface to an IP Address on the same subnet as the interface.

Visibility: Expert

10.146.4.3 **GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus**

Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.146.4.4 **GenApi::IInteger& DeviceCount**

Description: Number of compatible devices detected on current interface.

Visibility: Expert

#### 10.146.4.5 GenApi::IString& DeviceID

Description: [Interface](#) wide unique identifier of the selected device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 10.146.4.6 GenApi::IString& DeviceModelName

Description: Name of the device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 10.146.4.7 GenApi::Integer& DeviceSelector

Description: Selector for the different devices on this interface.

This value only changes on execution of "DeviceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Expert

#### 10.146.4.8 GenApi::IString& DeviceUnlock

Description: Unlocks devices for internal use.

Visibility: Expert

#### 10.146.4.9 GenApi::ICommand& DeviceUpdateList

Description: Updates the internal device list.

Visibility: Expert

#### 10.146.4.10 GenApi::IString& DeviceVendorName

Description: Name of the device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 10.146.4.11 GenApi::Integer& DevActionDeviceKey

Description: Key to authorize the action for the device.

Visibility: Expert

#### 10.146.4.12 **GenApi::Integer&GevActionGroupKey**

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

#### 10.146.4.13 **GenApi::Integer&GevActionGroupMask**

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

#### 10.146.4.14 **GenApi::Integer&GevActionTime**

Description: Provides the time in nanoseconds when the action is to be executed.

Visibility: Expert

#### 10.146.4.15 **GenApi::Integer&GevDeviceIPAddress**

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

#### 10.146.4.16 **GenApi::Integer&GevDeviceMACAddress**

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

#### 10.146.4.17 **GenApi::Integer&GevDeviceSubnetMask**

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

#### 10.146.4.18 **GenApi::Integer&GevInterfaceGateway**

Description: IP address of the selected gateway entry of this interface.

Visibility: Expert



**10.146.4.19 GenApi::Integer& GevInterfaceIPAddress**

Description: IP address of the selected subnet of this interface.

Visibility: Expert

**10.146.4.20 GenApi::Integer& GevInterfaceMACAddress**

Description: 48-bit MAC address of this interface.

Visibility: Expert

**10.146.4.21 GenApi::Integer& GevInterfaceSubnetMask**

Description: Subnet mask of the selected subnet of this interface.

Visibility: Expert

**10.146.4.22 GenApi::String& HostAdapterDriverVersion**

Description: Driver version of the interface's host adapter.

Visibility: Expert

**10.146.4.23 GenApi::String& HostAdapterName**

Description: User readable name of the interface's host adapter.

Visibility: Expert

**10.146.4.24 GenApi::String& HostAdapterVendor**

Description: User readable name of the host adapter's vendor.

Visibility: Expert

**10.146.4.25 GenApi::Integer& IncompatibleDeviceCount**

Description: Number of incompatible devices detected on current interface.

Visibility: Expert

**10.146.4.26 GenApi::String& IncompatibleDeviceID**

Description: [Interface](#) wide unique identifier of the selected incompatible device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

**10.146.4.27 GenApi::IString& IncompatibleDeviceModelName**

Description: Name of the incompatible device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

**10.146.4.28 GenApi::Integer& IncompatibleDeviceSelector**

Description: Selector for the devices that are not compatible with [Spinnaker](#) on this interface.

This value only changes on execution of "DeviceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Expert

**10.146.4.29 GenApi::IString& IncompatibleDeviceVendorName**

Description: Name of the incompatible device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

**10.146.4.30 GenApi::Integer& IncompatibleGevDeviceIPAddress**

Description: Current IP address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

**10.146.4.31 GenApi::Integer& IncompatibleGevDeviceMACAddress**

Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

**10.146.4.32 GenApi::Integer& IncompatibleGevDeviceSubnetMask**

Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

**10.146.4.33 GenApi::IString& InterfaceDisplayName**

Description: User readable name of the selected interface.

Visibility: Expert

## 10.146.4.34 GenApi::IString&amp; InterfaceID

Description: Transport layer Producer wide unique identifier of the selected interface.

Visibility: Expert

## 10.146.4.35 GenApi::IString&amp; InterfaceType

Description: Transport layer type of the interface.

Visibility: Expert

## 10.146.4.36 GenApi::IEnumerationT&lt;POEStatusEnum&gt;&amp; POEStatus

Description: Reports and controls the interface's power over Ethernet status.

Visibility: Expert

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

- [include/TransportLayerInterface.h](#)

## 10.147 TransportLayerStream Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

### Public Member Functions

- [TransportLayerStream](#) ([GenApi::INodeMap](#) \*nodeMapTLDevice)
- [~TransportLayerStream](#) ()

### Public Attributes

- [GenApi::IString](#) & [StreamID](#)  
*Description: Device unique ID for the data stream, e.g.*
- [GenApi::IEnumerationT](#) < [StreamTypeEnum](#) > & [StreamType](#)  
*Description: Stream type of the device.*
- [GenApi::Integer](#) & [StreamTotalBufferCount](#)  
*Description: Counts the number of image buffers that arrived since stream started.*
- [GenApi::Integer](#) & [StreamDefaultBufferCount](#)  
*Description: DEPRECATED; Replaced by [StreamBufferCountManual](#).*
- [GenApi::Integer](#) & [StreamDefaultBufferCountMax](#)  
*Description: DEPRECATED; Replaced by [StreamBufferCountMax](#).*
- [GenApi::IEnumerationT](#) < [StreamDefaultBufferCountModeEnum](#) > & [StreamDefaultBufferCountMode](#)  
*Description: DEPRECATED; Replaced by [StreamBufferCountMode](#).*
- [GenApi::Integer](#) & [StreamBufferCountManual](#)

- Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.*

  - [GenApi::Integer](#) & [StreamBufferCountResult](#)

*Description: Displays the number of buffers to be used on this stream upon acquisition start.*
- [GenApi::Integer](#) & [StreamBufferCountMax](#)

*Description: Controls the maximum number of buffers that should be used on this stream.*
- [GenApi::IEnumerationT](#) < [StreamBufferCountModeEnum](#) > & [StreamBufferCountMode](#)

*Description: Controls access to setting the number of buffers used for the stream.*
- [GenApi::IEnumerationT](#) < [StreamBufferHandlingModeEnum](#) > & [StreamBufferHandlingMode](#)

*Description: Available buffer handling modes of this data stream: Visibility: Beginner.*
- [GenApi::IBoolean](#) & [StreamCRCCheckEnable](#)

*Description: Enables or disables CRC checks on received images.*
- [GenApi::IBoolean](#) & [GevPacketResendMode](#)

*Description: Enables or disables the packet resend mechanism.*
- [GenApi::Integer](#) & [GevMaximumNumberResendRequests](#)

*Description: Maximum number of resend requests per image.*
- [GenApi::Integer](#) & [GevPacketResendTimeout](#)

*Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.*
- [GenApi::Integer](#) & [GevMaximumNumberResendBuffers](#)

*Description: This node is not used and has been deprecated.*
- [GenApi::Integer](#) & [GevTotalPacketCount](#)

*Description: Displays number of packets received on this stream.*
- [GenApi::Integer](#) & [GevFailedPacketCount](#)

*Description: Displays number of packets missed on this stream.*
- [GenApi::Integer](#) & [GevResendPacketCount](#)

*Description: Displays number of packets received after retransmit request on this stream.*
- [GenApi::Integer](#) & [StreamFailedBufferCount](#)

*Description: Displays number of incomplete images with missing leader/trailer information.*
- [GenApi::Integer](#) & [StreamBufferUnderrunCount](#)

*Description: Displays number of dropped images caused by driver running out of buffers.*
- [GenApi::Integer](#) & [GevResendRequestCount](#)

*Description: Displays number of packets requested to be retransmitted on this stream.*
- [GenApi::Integer](#) & [StreamBlockTransferSize](#)

*Description: Controls the image breakup size that should be used on this stream.*

## Protected Member Functions

- [TransportLayerStream](#) ()

## Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameraInternal](#)

## 10.147.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

## 10.147.2 Constructor & Destructor Documentation

10.147.2.1 **TransportLayerStream** ( **GenApi::INodeMap** \* *nodeMapTLDevice* )

10.147.2.2 **~TransportLayerStream** ( )

10.147.2.3 **TransportLayerStream** ( ) [protected]

## 10.147.3 Friends And Related Function Documentation

10.147.3.1 **friend class CameraBase** [friend]

10.147.3.2 **friend class CameraInternal** [friend]

10.147.3.3 **friend class ICameraBase** [friend]

## 10.147.4 Member Data Documentation

10.147.4.1 **GenApi::Integer& GevFailedPacketCount**

Description: Displays number of packets missed on this stream.

Visibility: Expert

10.147.4.2 **GenApi::Integer& GevMaximumNumberResendBuffers**

Description: This node is not used and has been deprecated.

Visibility: Invisible

10.147.4.3 **GenApi::Integer& GevMaximumNumberResendRequests**

Description: Maximum number of resend requests per image.

Each resend request consists of a span of consecutive packet IDs. Visibility: Expert

10.147.4.4 **GenApi::Boolean& GevPacketResendMode**

Description: Enables or disables the packet resend mechanism.

Visibility: Expert

10.147.4.5 **GenApi::Integer& GevPacketResendTimeout**

Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.

Visibility: Expert

**10.147.4.6 GenApi::Integer& GenResendPacketCount**

Description: Displays number of packets received after retransmit request on this stream.

Visibility: Expert

**10.147.4.7 GenApi::Integer& GenResendRequestCount**

Description: Displays number of packets requested to be retransmitted on this stream.

Visibility: Expert

**10.147.4.8 GenApi::Integer& GenTotalPacketCount**

Description: Displays number of packets received on this stream.

Visibility: Expert

**10.147.4.9 GenApi::Integer& StreamBlockTransferSize**

Description: Controls the image breakup size that should be used on this stream.

Visibility: Expert

**10.147.4.10 GenApi::Integer& StreamBufferCountManual**

Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.

Visibility: Expert

**10.147.4.11 GenApi::Integer& StreamBufferCountMax**

Description: Controls the maximum number of buffers that should be used on this stream.

This value is calculated based on the available system memory. Visibility: Expert

**10.147.4.12 GenApi::EnumerationT<StreamBufferCountModeEnum>& StreamBufferCountMode**

Description: Controls access to setting the number of buffers used for the stream.

Locked to Manual mode on 32-bit Windows due to memory constraints. Visibility: Expert

**10.147.4.13 GenApi::Integer& StreamBufferCountResult**

Description: Displays the number of buffers to be used on this stream upon acquisition start.

Recalculated on acquisition start if in auto mode. Visibility: Expert

**10.147.4.14 GenApi::IEnumerationT<StreamBufferHandlingModeEnum> & StreamBufferHandlingMode**

Description: Available buffer handling modes of this data stream: Visibility: Beginner.

**10.147.4.15 GenApi::IInteger & StreamBufferUnderrunCount**

Description: Displays number of dropped images caused by driver running out of buffers.

Visibility: Expert

**10.147.4.16 GenApi::IBoolean & StreamCRCCheckEnable**

Description: Enables or disables CRC checks on received images.

Visibility: Expert

**10.147.4.17 GenApi::IInteger & StreamDefaultBufferCount**

Description: DEPRECATED; Replaced by StreamBufferCountManual.

Controls the number of buffers to be used on this stream upon acquisition start when in manual mode. Visibility: Invisible

**10.147.4.18 GenApi::IInteger & StreamDefaultBufferCountMax**

Description: DEPRECATED; Replaced by StreamBufferCountMax.

Controls the maximum number of buffers that should be used on this stream. This value is calculated based on the available system memory. Visibility: Invisible

**10.147.4.19 GenApi::IEnumerationT<StreamDefaultBufferCountModeEnum> & StreamDefaultBufferCountMode**

Description: DEPRECATED; Replaced by StreamBufferCountMode.

Controls access to setting the number of buffers used for the stream. Locked to Manual mode on 32-bit Windows due to memory constraints. Visibility: Invisible

**10.147.4.20 GenApi::IInteger & StreamFailedBufferCount**

Description: Displays number of incomplete images with missing leader/trailer information.

Visibility: Expert

#### 10.147.4.21 GenApi::IString& StreamID

Description: Device unique ID for the data stream, e.g.

a GUID. Visibility: Expert

#### 10.147.4.22 GenApi::Integer& StreamTotalBufferCount

Description: Counts the number of image buffers that arrived since stream started.

Visibility: Expert

#### 10.147.4.23 GenApi::IEnumerationT<StreamTypeEnum>& StreamType

Description: Stream type of the device.

Visibility: Expert

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

- include/[TransportLayerStream.h](#)

### 10.148 U3V\_CHUNK\_TRAILER Struct Reference

header of a GVCP request packet

#### Public Attributes

- uint32\_t [ChunkID](#)
- uint32\_t [ChunkLength](#)

#### 10.148.1 Detailed Description

header of a GVCP request packet

#### 10.148.2 Member Data Documentation

##### 10.148.2.1 uint32\_t ChunkID

##### 10.148.2.2 uint32\_t ChunkLength

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

- include/SpinGenApi/[ChunkAdapterU3V.h](#)



## 10.149 U3V\_COMMAND\_HEADER Struct Reference

U3V/GenCP command header.

### Public Attributes

- uint32\_t [Prefix](#)
- uint16\_t [Flags](#)
- uint16\_t [CommandId](#)
- uint16\_t [Length](#)
- uint16\_t [ReqId](#)

### 10.149.1 Detailed Description

U3V/GenCP command header.

### 10.149.2 Member Data Documentation

10.149.2.1 uint16\_t CommandId

10.149.2.2 uint16\_t Flags

10.149.2.3 uint16\_t Length

10.149.2.4 uint32\_t Prefix

10.149.2.5 uint16\_t ReqId

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

- include/SpinGenApi/[EventAdapterU3V.h](#)

## 10.150 U3V\_EVENT\_DATA Struct Reference

U3V/GenCP EVENT\_CMD specific command data.

### Public Attributes

- uint16\_t [Reserved](#)
- uint16\_t [EventId](#)
- uint64\_t [Timestamp](#)

### 10.150.1 Detailed Description

U3V/GenCP EVENT\_CMD specific command data.

### 10.150.2 Member Data Documentation

10.150.2.1 `uint16_t` EventId

10.150.2.2 `uint16_t` Reserved

10.150.2.3 `uint64_t` Timestamp

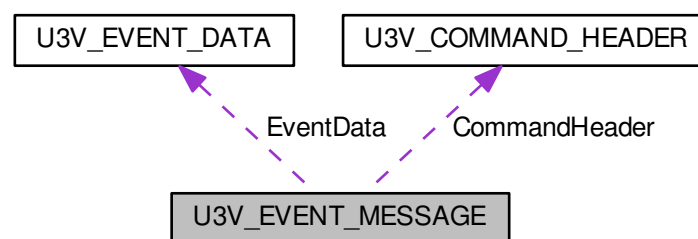
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterU3V.h`

## 10.151 U3V\_EVENT\_MESSAGE Struct Reference

Entire event data message (without the variable-sized data field)

Collaboration diagram for U3V\_EVENT\_MESSAGE:



### Public Attributes

- `U3V_COMMAND_HEADER` CommandHeader
- `U3V_EVENT_DATA` EventData

### 10.151.1 Detailed Description

Entire event data message (without the variable-sized data field)

### 10.151.2 Member Data Documentation

#### 10.151.2.1 U3V\_COMMAND\_HEADER CommandHeader

#### 10.151.2.2 U3V\_EVENT\_DATA EventData

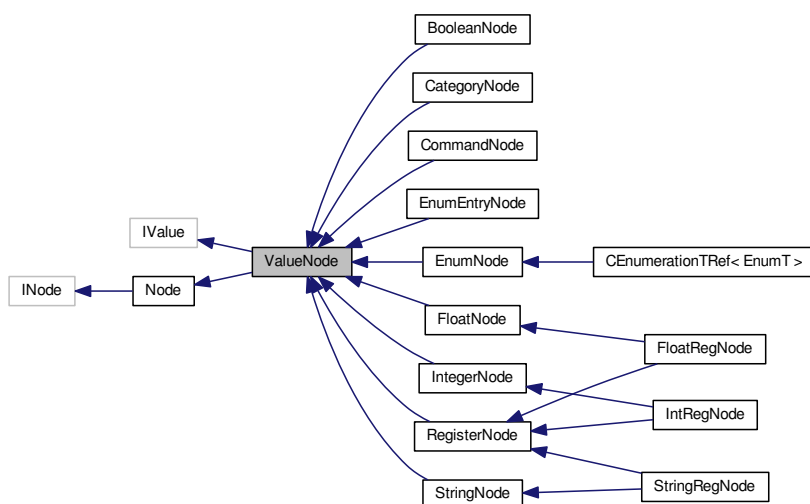
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

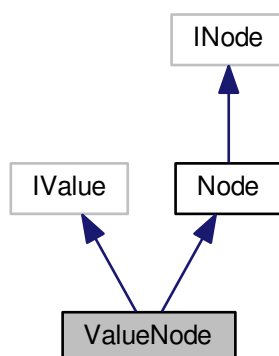
## 10.152 ValueNode Class Reference

[Interface](#) for value properties.

Inheritance diagram for ValueNode:



Collaboration diagram for ValueNode:



## Public Member Functions

- [ValueNode](#) ()  
*Constructor.*
- [ValueNode](#) (std::shared\_ptr< Node::NodeImpl > pValue)  
*constructor with [GenlCam](#) IValue*
- [~ValueNode](#) ()  
*Destructor.*
- virtual [INode](#) \* [GetNode](#) ()
- virtual [GenlCam::gcstring](#) [ToString](#) (bool [Verify](#)=false, bool IgnoreCache=false)  
*Get content of the node as string.*
- virtual void [FromString](#) (const [GenlCam::gcstring](#) &ValueStr, bool [Verify](#)=true)  
*Set content of the node as string.*
- virtual bool [IsValueCacheValid](#) () const  
*Checks if the value comes from cache or is requested from another node.*
- virtual void [SetReference](#) ([INode](#) \*pBase)  
*overload SetReference for Value*

## Additional Inherited Members

### 10.152.1 Detailed Description

[Interface](#) for value properties.

### 10.152.2 Constructor & Destructor Documentation

#### 10.152.2.1 [ValueNode](#) ( )

Constructor.

#### 10.152.2.2 [ValueNode](#) ( std::shared\_ptr< Node::NodeImpl > pValue )

constructor with [GenlCam](#) IValue

#### 10.152.2.3 [~ValueNode](#) ( )

Destructor.

### 10.152.3 Member Function Documentation

#### 10.152.3.1 virtual void [FromString](#) ( const [GenlCam::gcstring](#) & [ValueStr](#), bool [Verify](#)=true ) [virtual]

Set content of the node as string.

## Parameters

|                 |                                                            |
|-----------------|------------------------------------------------------------|
| <i>ValueStr</i> | The value to set                                           |
| <i>Verify</i>   | Enables AccessMode and Range verification (default = true) |

10.152.3.2 `virtual INode* GetNode ( )` [virtual]

10.152.3.3 `virtual bool IsValueCacheValid ( ) const` [virtual]

Checks if the value comes from cache or is requested from another node.

10.152.3.4 `virtual void SetReference ( INode * pBase )` [virtual]

overload SetReference for Value

Reimplemented from [Node](#).

Reimplemented in [FloatNode](#), [IntegerNode](#), [EnumNode](#), [CEnumerationTRef< EnumT >](#), [StringNode](#), [RegisterNode](#), [BooleanNode](#), [CommandNode](#), [EnumEntryNode](#), [CategoryNode](#), [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

10.152.3.5 `virtual GenICam::gcstring ToString ( bool Verify = false, bool IgnoreCache = false )` [virtual]

Get content of the node as string.

## Parameters

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| <i>Verify</i>      | Enables Range verification (default = false). The AccessMode is always checked |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false)                |

## Returns

The value read

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

- [include/SpinGenApi/ValueNode.h](#)

## 10.153 Version\_t Struct Reference

Version.

## Public Attributes

- uint16\_t [Major](#)
- uint16\_t [Minor](#)  
*a is incompatible with b if  $a \neq b$*
- uint16\_t [SubMinor](#)  
*a is incompatible b  $a > b$*

### 10.153.1 Detailed Description

Version.

### 10.153.2 Member Data Documentation

#### 10.153.2.1 uint16\_t Major

#### 10.153.2.2 uint16\_t Minor

a is incompatible with b if  $a \neq b$

#### 10.153.2.3 uint16\_t SubMinor

a is incompatible b  $a > b$

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

- include/SpinGenApi/[GCTypes.h](#)

## Chapter 11

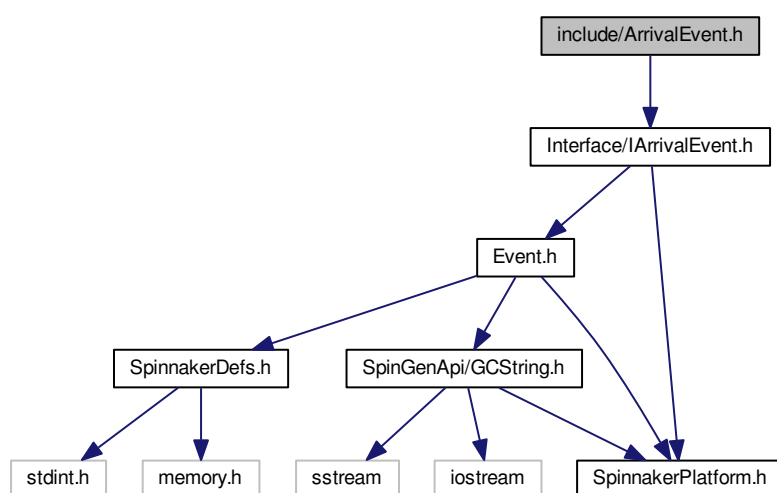
# File Documentation

### 11.1 doc/Doxygen/spindocs/Licensing.dox File Reference

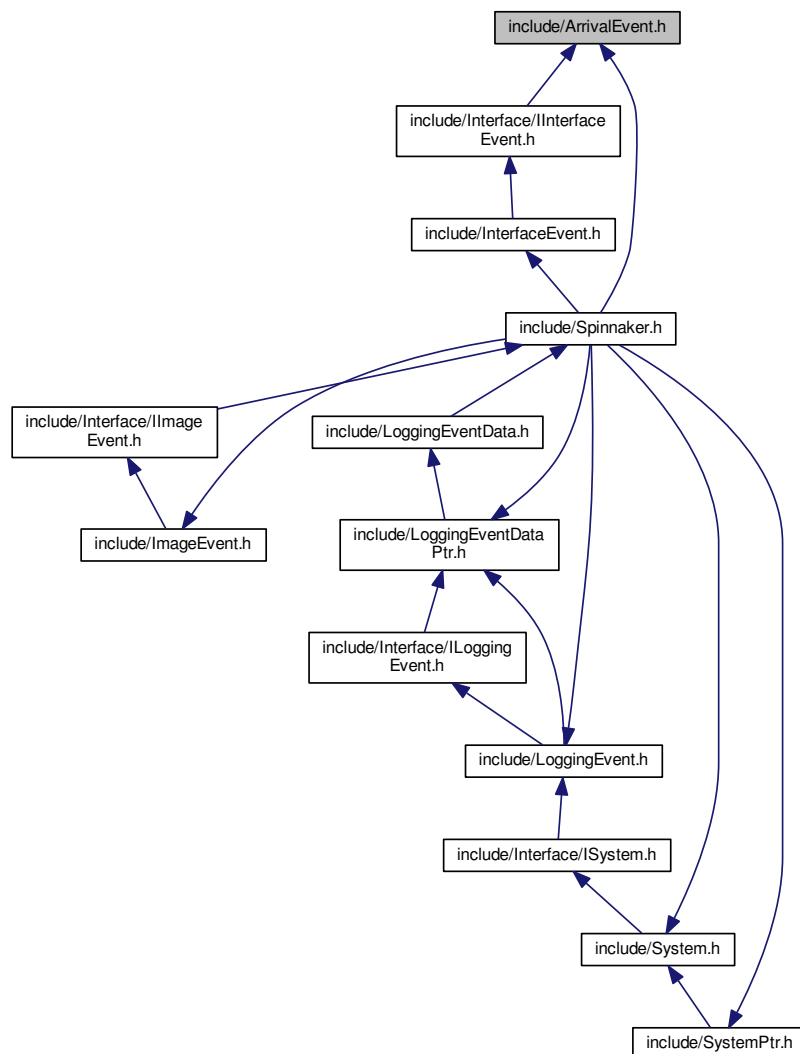
### 11.2 doc/Doxygen/spindocs/MainPage.dox File Reference

### 11.3 include/ArrivalEvent.h File Reference

Include dependency graph for ArrivalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ArrivalEvent](#)

*An event handler for capturing the device arrival event.*

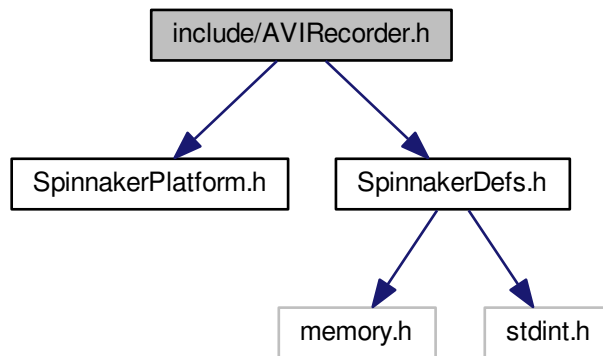
## Namespaces

- [Spinnaker](#)



## 11.4 include/AVIRecorder.h File Reference

Include dependency graph for AVIRecorder.h:



### Namespaces

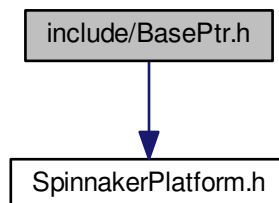
- [Spinnaker](#)

### Functions

- class [DEPRECATED\\_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER\_API `AVIRecorder`  
*Provides the functionality for the user to record images to an AVI file.*

## 11.5 include/BasePtr.h File Reference

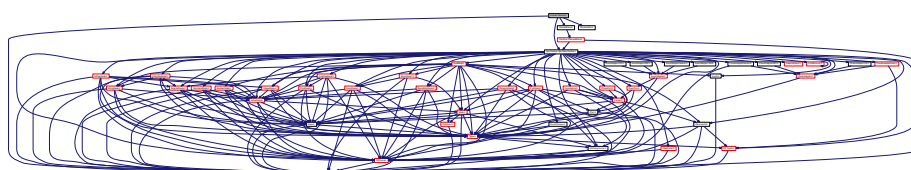
Include dependency graph for BasePtr.h:



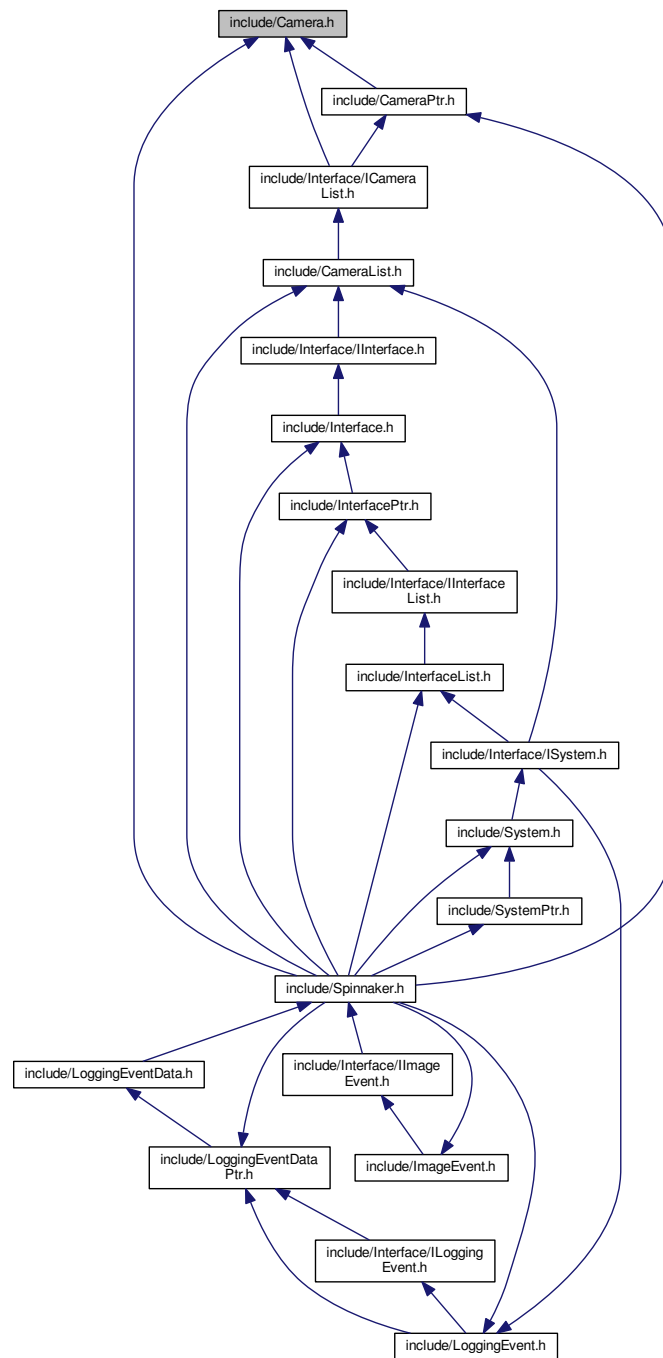


## 11.6 include/Camera.h File Reference

Include dependency graph for Camera.h:



This graph shows which files directly or indirectly include this file:



## Classes

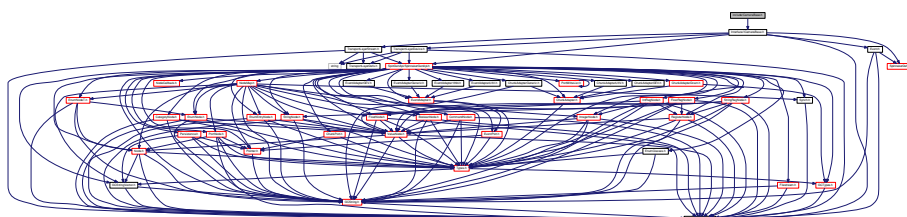
- class [Camera](#)  
The camera object class.

## Namespaces

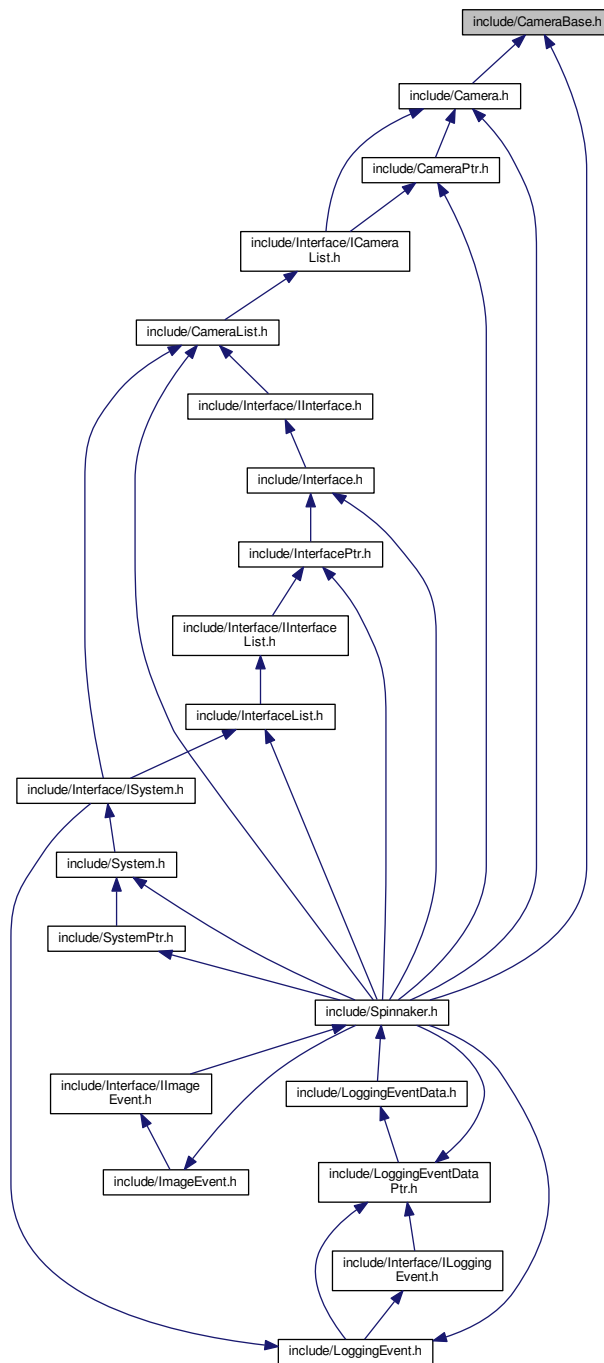
- [Spinnaker](#)

## 11.7 include/CameraBase.h File Reference

Include dependency graph for CameraBase.h:



This graph shows which files directly or indirectly include this file:



## Classes

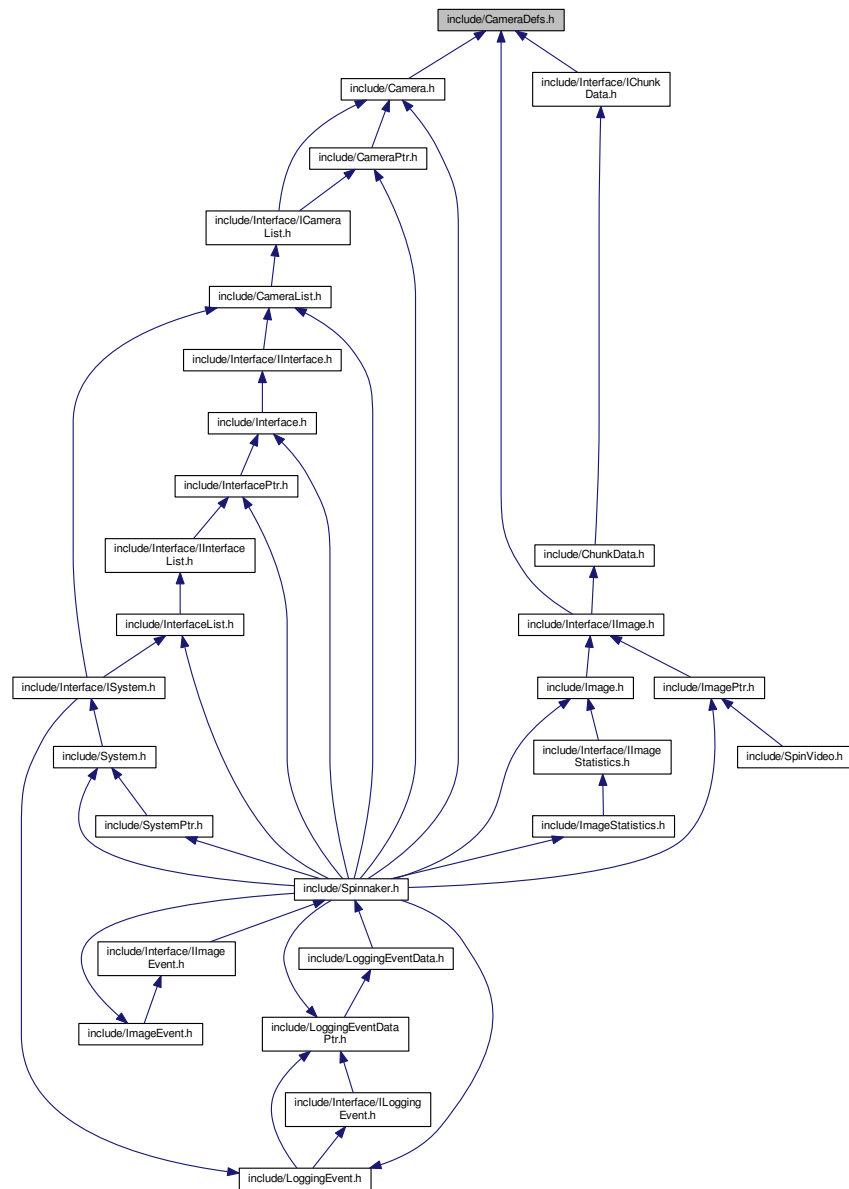
- class [CameraBase](#)  
The base class for the camera object.

## Namespaces

- [Spinnaker](#)

## 11.8 include/CameraDefs.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Spinnaker](#)

### Enumerations

- enum [LUTSelectorEnums](#) {  
[LUTSelector\\_LUT1](#),  
[NUM\\_LUTSELECTOR](#) }

*The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.*

- enum ExposureModeEnums {  
ExposureMode\_Timed,  
ExposureMode\_TriggerWidth,  
NUM\_EXPOSUREMODE }
- enum AcquisitionModeEnums {  
AcquisitionMode\_Continuous,  
AcquisitionMode\_SingleFrame,  
AcquisitionMode\_MultiFrame,  
NUM\_ACQUISITIONMODE }
- enum TriggerSourceEnums {  
TriggerSource\_Software,  
TriggerSource\_Line0,  
TriggerSource\_Line1,  
TriggerSource\_Line2,  
TriggerSource\_Line3,  
TriggerSource\_UserOutput0,  
TriggerSource\_UserOutput1,  
TriggerSource\_UserOutput2,  
TriggerSource\_UserOutput3,  
TriggerSource\_Counter0Start,  
TriggerSource\_Counter1Start,  
TriggerSource\_Counter0End,  
TriggerSource\_Counter1End,  
TriggerSource\_LogicBlock0,  
TriggerSource\_LogicBlock1,  
TriggerSource\_Action0,  
NUM\_TRIGGERSOURCE }
- enum TriggerActivationEnums {  
TriggerActivation\_LevelLow,  
TriggerActivation\_LevelHigh,  
TriggerActivation\_FallingEdge,  
TriggerActivation\_RisingEdge,  
TriggerActivation\_AnyEdge,  
NUM\_TRIGGERACTIVATION }
- enum SensorShutterModeEnums {  
SensorShutterMode\_Global,  
SensorShutterMode\_Rolling,  
SensorShutterMode\_GlobalReset,  
NUM\_SENSORSHUTTERMODE }
- enum TriggerModeEnums {  
TriggerMode\_Off,  
TriggerMode\_On,  
NUM\_TRIGGERMODE }
- enum TriggerOverlapEnums {  
TriggerOverlap\_Off,  
TriggerOverlap\_ReadOut,  
TriggerOverlap\_PreviousFrame,  
NUM\_TRIGGEROVERLAP }
- enum TriggerSelectorEnums {  
TriggerSelector\_AcquisitionStart,  
TriggerSelector\_FrameStart,  
TriggerSelector\_FrameBurstStart,  
NUM\_TRIGGERSELECTOR }
- enum ExposureAutoEnums {  
ExposureAuto\_Off,  
ExposureAuto\_Once,  
ExposureAuto\_Continuous,



- ```
NUM_EXPOSUREAUTO }
```
- enum EventSelectorEnums {  
EventSelector\_Error,  
EventSelector\_ExposureEnd,  
EventSelector\_SerialPortReceive,  
NUM\_EVENTSELECTOR }
  - enum EventNotificationEnums {  
EventNotification\_On,  
EventNotification\_Off,  
NUM\_EVENTNOTIFICATION }
  - enum LogicBlockSelectorEnums {  
LogicBlockSelector\_LogicBlock0,  
LogicBlockSelector\_LogicBlock1,  
NUM\_LOGICBLOCKSELECTOR }
  - enum LogicBlockLUTInputActivationEnums {  
LogicBlockLUTInputActivation\_LevelLow,  
LogicBlockLUTInputActivation\_LevelHigh,  
LogicBlockLUTInputActivation\_FallingEdge,  
LogicBlockLUTInputActivation\_RisingEdge,  
LogicBlockLUTInputActivation\_AnyEdge,  
NUM\_LOGICBLOCKLUTINPUTACTIVATION }
  - enum LogicBlockLUTInputSelectorEnums {  
LogicBlockLUTInputSelector\_Input0,  
LogicBlockLUTInputSelector\_Input1,  
LogicBlockLUTInputSelector\_Input2,  
LogicBlockLUTInputSelector\_Input3,  
NUM\_LOGICBLOCKLUTINPUTSELECTOR }
  - enum LogicBlockLUTInputSourceEnums {  
LogicBlockLUTInputSource\_Zero,  
LogicBlockLUTInputSource\_Line0,  
LogicBlockLUTInputSource\_Line1,  
LogicBlockLUTInputSource\_Line2,  
LogicBlockLUTInputSource\_Line3,  
LogicBlockLUTInputSource\_UserOutput0,  
LogicBlockLUTInputSource\_UserOutput1,  
LogicBlockLUTInputSource\_UserOutput2,  
LogicBlockLUTInputSource\_UserOutput3,  
LogicBlockLUTInputSource\_Counter0Start,  
LogicBlockLUTInputSource\_Counter1Start,  
LogicBlockLUTInputSource\_Counter0End,  
LogicBlockLUTInputSource\_Counter1End,  
LogicBlockLUTInputSource\_LogicBlock0,  
LogicBlockLUTInputSource\_LogicBlock1,  
LogicBlockLUTInputSource\_ExposureStart,  
LogicBlockLUTInputSource\_ExposureEnd,  
LogicBlockLUTInputSource\_FrameTriggerWait,  
LogicBlockLUTInputSource\_AcquisitionActive,  
NUM\_LOGICBLOCKLUTINPUTSOURCE }
  - enum LogicBlockLUTSelectorEnums {  
LogicBlockLUTSelector\_Value,  
LogicBlockLUTSelector\_Enable,  
NUM\_LOGICBLOCKLUTSELECTOR }
  - enum ColorTransformationSelectorEnums {  
ColorTransformationSelector\_RGBtoRGB,  
ColorTransformationSelector\_RGBtoYUV,  
NUM\_COLORTRANSFORMATIONSELECTOR }
  - enum RgbTransformLightSourceEnums {

```

    RgbTransformLightSource_General,
    RgbTransformLightSource_Tungsten2800K,
    RgbTransformLightSource_WarmFluorescent3000K,
    RgbTransformLightSource_CoolFluorescent4000K,
    RgbTransformLightSource_Daylight5000K,
    RgbTransformLightSource_Cloudy6500K,
    RgbTransformLightSource_Shade8000K,
    RgbTransformLightSource_Custom,
    NUM_RGBTRANSFORMLIGHTSOURCE }

• enum ColorTransformationValueSelectorEnums {
    ColorTransformationValueSelector_Gain00,
    ColorTransformationValueSelector_Gain01,
    ColorTransformationValueSelector_Gain02,
    ColorTransformationValueSelector_Gain10,
    ColorTransformationValueSelector_Gain11,
    ColorTransformationValueSelector_Gain12,
    ColorTransformationValueSelector_Gain20,
    ColorTransformationValueSelector_Gain21,
    ColorTransformationValueSelector_Gain22,
    ColorTransformationValueSelector_Offset0,
    ColorTransformationValueSelector_Offset1,
    ColorTransformationValueSelector_Offset2,
    NUM_COLORTRANSFORMATIONVALUESELECTOR }

• enum DeviceRegistersEndiannessEnums {
    DeviceRegistersEndianness_Little,
    DeviceRegistersEndianness_Big,
    NUM_DEVICEREGISTERSENDIANNES }

• enum DeviceScanTypeEnums {
    DeviceScanType_Areascan,
    NUM_DEVICESCANTYPE }

• enum DeviceCharacterSetEnums {
    DeviceCharacterSet_UTF8,
    DeviceCharacterSet_ASCII,
    NUM_DEVICECHARACTERSET }

• enum DeviceTLTypeEnums {
    DeviceTLType_GigEVision,
    DeviceTLType_CameraLink,
    DeviceTLType_CameraLinkHS,
    DeviceTLType_CoaXPress,
    DeviceTLType_USB3Vision,
    DeviceTLType_Custom,
    NUM_DEVICETLTYPE }

• enum DevicePowerSupplySelectorEnums {
    DevicePowerSupplySelector_External,
    NUM_DEVICEPOWERSUPPLYSELECTOR }

• enum DeviceTemperatureSelectorEnums {
    DeviceTemperatureSelector_Sensor,
    NUM_DEVICETEMPERATURESELECTOR }

• enum DeviceIndicatorModeEnums {
    DeviceIndicatorMode_Inactive,
    DeviceIndicatorMode_Active,
    DeviceIndicatorMode_ErrorStatus,
    NUM_DEVICEINDICATORMODE }

• enum AutoExposureControlPriorityEnums {
    AutoExposureControlPriority_Gain,
    AutoExposureControlPriority_ExposureTime,
    NUM_AUTOEXPOSURECONTROLPRIORITY }

```

- enum [AutoExposureMeteringModeEnums](#) {  
    AutoExposureMeteringMode\_Average,  
    AutoExposureMeteringMode\_Spot,  
    AutoExposureMeteringMode\_Partial,  
    AutoExposureMeteringMode\_CenterWeighted,  
    AutoExposureMeteringMode\_HistogramPeak,  
    NUM\_AUTOEXPOSUREMETERINGMODE }
- enum [BalanceWhiteAutoProfileEnums](#) {  
    BalanceWhiteAutoProfile\_Indoor,  
    BalanceWhiteAutoProfile\_Outdoor,  
    NUM\_BALANCEWHITEAUTOPROFILE }
- enum [AutoAlgorithmSelectorEnums](#) {  
    AutoAlgorithmSelector\_Awb,  
    AutoAlgorithmSelector\_Ae,  
    NUM\_AUTOALGORITHMSELECTOR }
- enum [AutoExposureTargetGreyValueAutoEnums](#) {  
    AutoExposureTargetGreyValueAuto\_Off,  
    AutoExposureTargetGreyValueAuto\_Continuous,  
    NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO }
- enum [AutoExposureLightingModeEnums](#) {  
    AutoExposureLightingMode\_AutoDetect,  
    AutoExposureLightingMode\_Backlight,  
    AutoExposureLightingMode\_Frontlight,  
    AutoExposureLightingMode\_Normal,  
    NUM\_AUTOEXPOSURELIGHTINGMODE }
- enum [GevIEEE1588StatusEnums](#) {  
    GevIEEE1588Status\_Initializing,  
    GevIEEE1588Status\_Faulty,  
    GevIEEE1588Status\_Disabled,  
    GevIEEE1588Status\_Listening,  
    GevIEEE1588Status\_PreMaster,  
    GevIEEE1588Status\_Master,  
    GevIEEE1588Status\_Passive,  
    GevIEEE1588Status\_Uncalibrated,  
    GevIEEE1588Status\_Slave,  
    NUM\_GEVIEEE1588STATUS }
- enum [GevIEEE1588ModeEnums](#) {  
    GevIEEE1588Mode\_Auto,  
    GevIEEE1588Mode\_SlaveOnly,  
    NUM\_GEVIEEE1588MODE }
- enum [GevIEEE1588ClockAccuracyEnums](#) {  
    GevIEEE1588ClockAccuracy\_Unknown,  
    NUM\_GEVIEEE1588CLOCKACCURACY }
- enum [GevCCPEnums](#) {  
    GevCCP\_OpenAccess,  
    GevCCP\_ExclusiveAccess,  
    GevCCP\_ControlAccess,  
    NUM\_GEVCCP }
- enum [GevSupportedOptionSelectorEnums](#) {

```

GevSupportedOptionSelector_UserDefinedName,
GevSupportedOptionSelector_SerialNumber,
GevSupportedOptionSelector_HeartbeatDisable,
GevSupportedOptionSelector_LinkSpeed,
GevSupportedOptionSelector_CCPApplicationSocket,
GevSupportedOptionSelector_ManifestTable,
GevSupportedOptionSelector_TestData,
GevSupportedOptionSelector_DiscoveryAckDelay,
GevSupportedOptionSelector_DiscoveryAckDelayWritable,
GevSupportedOptionSelector_ExtendedStatusCodes,
GevSupportedOptionSelector_Action,
GevSupportedOptionSelector_PendingAck,
GevSupportedOptionSelector_EventData,
GevSupportedOptionSelector_Event,
GevSupportedOptionSelector_PacketResend,
GevSupportedOptionSelector_WriteMem,
GevSupportedOptionSelector_CommandsConcatenation,
GevSupportedOptionSelector_IPConfigurationLLA,
GevSupportedOptionSelector_IPConfigurationDHCP,
GevSupportedOptionSelector_IPConfigurationPersistentIP,
GevSupportedOptionSelector_StreamChannelSourceSocket,
GevSupportedOptionSelector_MessageChannelSourceSocket,
NUM_GEVSUPPORTEDOPTIONSELECTOR }

• enum BlackLevelSelectorEnums {
    BlackLevelSelector_All,
    BlackLevelSelector_Analog,
    BlackLevelSelector_Digital,
    NUM_BLACKLEVELSELECTOR }

• enum BalanceWhiteAutoEnums {
    BalanceWhiteAuto_Off,
    BalanceWhiteAuto_Once,
    BalanceWhiteAuto_Continuous,
    NUM_BALANCEWHITEAUTO }

• enum GainAutoEnums {
    GainAuto_Off,
    GainAuto_Once,
    GainAuto_Continuous,
    NUM_GAINAUTO }

• enum BalanceRatioSelectorEnums {
    BalanceRatioSelector_Red,
    BalanceRatioSelector_Blue,
    NUM_BALANCERATIOSELECTOR }

• enum GainSelectorEnums {
    GainSelector_All,
    NUM_GAINSELECTOR }

• enum DefectCorrectionModeEnums {
    DefectCorrectionMode_Average,
    DefectCorrectionMode_Highlight,
    DefectCorrectionMode_Zero,
    NUM_DEFECTCORRECTIONMODE }

• enum UserSetSelectorEnums {
    UserSetSelector_Default,
    UserSetSelector_UserSet0,
    UserSetSelector_UserSet1,
    NUM_USERSETSELECTOR }

• enum UserSetDefaultEnums {

```

- UserSetDefault\_Default,  
UserSetDefault\_UserSet0,  
UserSetDefault\_UserSet1,  
NUM\_USERSETDEFAULT }
- enum SerialPortBaudRateEnums {  
SerialPortBaudRate\_Baud300,  
SerialPortBaudRate\_Baud600,  
SerialPortBaudRate\_Baud1200,  
SerialPortBaudRate\_Baud2400,  
SerialPortBaudRate\_Baud4800,  
SerialPortBaudRate\_Baud9600,  
SerialPortBaudRate\_Baud14400,  
SerialPortBaudRate\_Baud19200,  
SerialPortBaudRate\_Baud38400,  
SerialPortBaudRate\_Baud57600,  
SerialPortBaudRate\_Baud115200,  
SerialPortBaudRate\_Baud230400,  
SerialPortBaudRate\_Baud460800,  
SerialPortBaudRate\_Baud921600,  
NUM\_SERIALPORTBAUDRATE }
- enum SerialPortParityEnums {  
SerialPortParity\_None,  
SerialPortParity\_Odd,  
SerialPortParity\_Even,  
SerialPortParity\_Mark,  
SerialPortParity\_Space,  
NUM\_SERIALPORTPARITY }
- enum SerialPortSelectorEnums {  
SerialPortSelector\_SerialPort0,  
NUM\_SERIALPORTSELECTOR }
- enum SerialPortStopBitsEnums {  
SerialPortStopBits\_Bits1,  
SerialPortStopBits\_Bits1AndAHalf,  
SerialPortStopBits\_Bits2,  
NUM\_SERIALPORTSTOPBITS }
- enum SerialPortSourceEnums {  
SerialPortSource\_Line0,  
SerialPortSource\_Line1,  
SerialPortSource\_Line2,  
SerialPortSource\_Line3,  
SerialPortSource\_Off,  
NUM\_SERIALPORTSOURCE }
- enum SequencerModeEnums {  
SequencerMode\_Off,  
SequencerMode\_On,  
NUM\_SEQUENCERMODE }
- enum SequencerConfigurationValidEnums {  
SequencerConfigurationValid\_No,  
SequencerConfigurationValid\_Yes,  
NUM\_SEQUENCERCONFIGURATIONVALID }
- enum SequencerSetValidEnums {  
SequencerSetValid\_No,  
SequencerSetValid\_Yes,  
NUM\_SEQUENCERSETVALID }
- enum SequencerTriggerActivationEnums {

- ```

SequencerTriggerActivation_RisingEdge,
SequencerTriggerActivation_FallingEdge,
SequencerTriggerActivation_AnyEdge,
SequencerTriggerActivation_LevelHigh,
SequencerTriggerActivation_LevelLow,
NUM_SEQUENCERTRIGGERACTIVATION }

```
- enum SequencerConfigurationModeEnums {

```

SequencerConfigurationMode_Off,
SequencerConfigurationMode_On,
NUM_SEQUENCERCONFIGURATIONMODE }

```
  - enum SequencerTriggerSourceEnums {

```

SequencerTriggerSource_Off,
SequencerTriggerSource_FrameStart,
NUM_SEQUENCERTRIGGERSOURCE }

```
  - enum TransferQueueModeEnums {

```

TransferQueueMode_FirstInFirstOut,
NUM_TRANSFERQUEUEMODE }

```
  - enum TransferOperationModeEnums {

```

TransferOperationMode_Continuous,
TransferOperationMode_MultiBlock,
NUM_TRANSFEROPERATIONMODE }

```
  - enum TransferControlModeEnums {

```

TransferControlMode_Basic,
TransferControlMode_Automatic,
TransferControlMode_UserControlled,
NUM_TRANSFERCONTROLMODE }

```
  - enum ChunkGainSelectorEnums {

```

ChunkGainSelector_All,
ChunkGainSelector_Red,
ChunkGainSelector_Green,
ChunkGainSelector_Blue,
NUM_CHUNKGAINSELECTOR }

```
  - enum ChunkSelectorEnums {

```

ChunkSelector_Image,
ChunkSelector_CRC,
ChunkSelector_FrameID,
ChunkSelector_OffsetX,
ChunkSelector_OffsetY,
ChunkSelector_Width,
ChunkSelector_Height,
ChunkSelector_ExposureTime,
ChunkSelector_Gain,
ChunkSelector_BlackLevel,
ChunkSelector_PixelFormat,
ChunkSelector_Timestamp,
ChunkSelector_SequencerSetActive,
ChunkSelector_SerialData,
ChunkSelector_ExposureEndLineStatusAll,
NUM_CHUNKSELECTOR }

```
  - enum ChunkBlackLevelSelectorEnums {

```

ChunkBlackLevelSelector_All,
NUM_CHUNKBLACKLEVELSELECTOR }

```
  - enum ChunkPixelFormatEnums {

- ChunkPixelFormat\_Mono8,
- ChunkPixelFormat\_Mono12Packed,
- ChunkPixelFormat\_Mono16,
- ChunkPixelFormat\_RGB8Packed,
- ChunkPixelFormat\_YUV422Packed,
- ChunkPixelFormat\_BayerGR8,
- ChunkPixelFormat\_BayerRG8,
- ChunkPixelFormat\_BayerGB8,
- ChunkPixelFormat\_BayerBG8,
- ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY,
- NUM\_CHUNKPIXELFORMAT }
- enum FileOperationStatusEnums {
  - FileOperationStatus\_Success,
  - FileOperationStatus\_Failure,
  - FileOperationStatus\_Overflow,
  - NUM\_FILEOPERATIONSTATUS }
- enum FileOpenModeEnums {
  - FileOpenMode\_Read,
  - FileOpenMode\_Write,
  - FileOpenMode\_ReadWrite,
  - NUM\_FILEOPENMODE }
- enum FileOperationSelectorEnums {
  - FileOperationSelector\_Open,
  - FileOperationSelector\_Close,
  - FileOperationSelector\_Read,
  - FileOperationSelector\_Write,
  - FileOperationSelector\_Delete,
  - NUM\_FILEOPERATIONSELECTOR }
- enum FileSelectorEnums {
  - FileSelector\_UserSetDefault,
  - FileSelector\_UserSet0,
  - FileSelector\_UserSet1,
  - FileSelector\_UserFile1,
  - FileSelector\_SerialPort0,
  - NUM\_FILESELECTOR }
- enum BinningSelectorEnums {
  - BinningSelector\_All,
  - BinningSelector\_Sensor,
  - BinningSelector\_ISP,
  - NUM\_BINNINGSELECTOR }
- enum TestPatternGeneratorSelectorEnums {
  - TestPatternGeneratorSelector\_Sensor,
  - TestPatternGeneratorSelector\_PipelineStart,
  - NUM\_TESTPATTERNGENERATORSELECTOR }
- enum TestPatternEnums {
  - TestPattern\_Off,
  - TestPattern\_Increment,
  - TestPattern\_SensorTestPattern,
  - NUM\_TESTPATTERN }
- enum PixelColorFilterEnums {
  - PixelColorFilter\_None,
  - PixelColorFilter\_BayerRG,
  - PixelColorFilter\_BayerGB,
  - PixelColorFilter\_BayerGR,
  - PixelColorFilter\_BayerBG,
  - NUM\_PIXELCOLORFILTER }
- enum AdcBitDepthEnums {

```
AdcBitDepth_Bit8,  
AdcBitDepth_Bit10,  
AdcBitDepth_Bit12,  
AdcBitDepth_Bit14,  
NUM_ADCBITDEPTH }
```

- enum DecimationHorizontalModeEnums {  
DecimationHorizontalMode\_Discard,  
NUM\_DECIMATIONHORIZONTALMODE }

- enum BinningVerticalModeEnums {  
BinningVerticalMode\_Sum,  
BinningVerticalMode\_Average,  
NUM\_BINNINGVERTICALMODE }

- enum PixelSizeEnums {  
PixelSize\_Bpp1,  
PixelSize\_Bpp2,  
PixelSize\_Bpp4,  
PixelSize\_Bpp8,  
PixelSize\_Bpp10,  
PixelSize\_Bpp12,  
PixelSize\_Bpp14,  
PixelSize\_Bpp16,  
PixelSize\_Bpp20,  
PixelSize\_Bpp24,  
PixelSize\_Bpp30,  
PixelSize\_Bpp32,  
PixelSize\_Bpp36,  
PixelSize\_Bpp48,  
PixelSize\_Bpp64,  
PixelSize\_Bpp96,  
NUM\_PIXELSIZE }

- enum DecimationSelectorEnums {  
DecimationSelector\_All,  
DecimationSelector\_Sensor,  
NUM\_DECIMATIONSELECTOR }

- enum ImageCompressionModeEnums {  
ImageCompressionMode\_Off,  
ImageCompressionMode\_Lossless,  
NUM\_IMAGECOMPRESSIONMODE }

- enum BinningHorizontalModeEnums {  
BinningHorizontalMode\_Sum,  
BinningHorizontalMode\_Average,  
NUM\_BINNINGHORIZONTALMODE }

- enum PixelFormatEnums {



PixelFormat\_Mono8,  
PixelFormat\_Mono16,  
PixelFormat\_RGB8Packed,  
PixelFormat\_BayerGR8,  
PixelFormat\_BayerRG8,  
PixelFormat\_BayerGB8,  
PixelFormat\_BayerBG8,  
PixelFormat\_BayerGR16,  
PixelFormat\_BayerRG16,  
PixelFormat\_BayerGB16,  
PixelFormat\_BayerBG16,  
PixelFormat\_Mono12Packed,  
PixelFormat\_BayerGR12Packed,  
PixelFormat\_BayerRG12Packed,  
PixelFormat\_BayerGB12Packed,  
PixelFormat\_BayerBG12Packed,  
PixelFormat\_YUV411Packed,  
PixelFormat\_YUV422Packed,  
PixelFormat\_YUV444Packed,  
PixelFormat\_Mono12p,  
PixelFormat\_BayerGR12p,  
PixelFormat\_BayerRG12p,  
PixelFormat\_BayerGB12p,  
PixelFormat\_BayerBG12p,  
PixelFormat\_YCbCr8,  
PixelFormat\_YCbCr422\_8,  
PixelFormat\_YCbCr411\_8,  
PixelFormat\_BGR8,  
PixelFormat\_BGRa8,  
PixelFormat\_Mono10Packed,  
PixelFormat\_BayerGR10Packed,  
PixelFormat\_BayerRG10Packed,  
PixelFormat\_BayerGB10Packed,  
PixelFormat\_BayerBG10Packed,  
PixelFormat\_Mono10p,  
PixelFormat\_BayerGR10p,  
PixelFormat\_BayerRG10p,  
PixelFormat\_BayerGB10p,  
PixelFormat\_BayerBG10p,  
PixelFormat\_Mono1p,  
PixelFormat\_Mono2p,  
PixelFormat\_Mono4p,  
PixelFormat\_Mono8s,  
PixelFormat\_Mono10,  
PixelFormat\_Mono12,  
PixelFormat\_Mono14,  
PixelFormat\_BayerBG10,  
PixelFormat\_BayerBG12,  
PixelFormat\_BayerGB10,  
PixelFormat\_BayerGB12,  
PixelFormat\_BayerGR10,  
PixelFormat\_BayerGR12,  
PixelFormat\_BayerRG10,  
PixelFormat\_BayerRG12,  
PixelFormat\_RGBa8,  
PixelFormat\_RGBa10,  
PixelFormat\_RGBa10p,  
PixelFormat\_RGBa12,  
PixelFormat\_RGBa12p,  
PixelFormat\_RGBa14,  
PixelFormat\_RGBa16,  
PixelFormat\_RGB8,  
PixelFormat\_RGB8\_Planar,  
PixelFormat\_RGB10,  
PixelFormat\_RGB10\_Planar,

- NUM\_PIXELFORMAT }
- enum DecimationVerticalModeEnums {  
DecimationVerticalMode\_Discard,  
NUM\_DECIMATIONVERTICALMODE }
- enum LineModeEnums {  
LineMode\_Input,  
LineMode\_Output,  
NUM\_LINEMODE }
- enum LineSourceEnums {  
LineSource\_Off,  
LineSource\_Line0,  
LineSource\_Line1,  
LineSource\_Line2,  
LineSource\_Line3,  
LineSource\_UserOutput0,  
LineSource\_UserOutput1,  
LineSource\_UserOutput2,  
LineSource\_UserOutput3,  
LineSource\_Counter0Active,  
LineSource\_Counter1Active,  
LineSource\_LogicBlock0,  
LineSource\_LogicBlock1,  
LineSource\_ExposureActive,  
LineSource\_FrameTriggerWait,  
LineSource\_SerialPort0,  
LineSource\_PPSSignal,  
LineSource\_AllPixel,  
LineSource\_AnyPixel,  
NUM\_LINESOURCE }
- enum LineInputFilterSelectorEnums {  
LineInputFilterSelector\_Deglintch,  
LineInputFilterSelector\_Debounce,  
NUM\_LINEINPUTFILTERSELECTOR }
- enum UserOutputSelectorEnums {  
UserOutputSelector\_UserOutput0,  
UserOutputSelector\_UserOutput1,  
UserOutputSelector\_UserOutput2,  
UserOutputSelector\_UserOutput3,  
NUM\_USEROUTPUTSELECTOR }
- enum LineFormatEnums {  
LineFormat\_NoConnect,  
LineFormat\_TriState,  
LineFormat\_TTL,  
LineFormat\_LVDS,  
LineFormat\_RS422,  
LineFormat\_OptoCoupled,  
LineFormat\_OpenDrain,  
NUM\_LINEFORMAT }
- enum LineSelectorEnums {  
LineSelector\_Line0,  
LineSelector\_Line1,  
LineSelector\_Line2,  
LineSelector\_Line3,  
NUM\_LINESELECTOR }
- enum ExposureActiveModeEnums {  
ExposureActiveMode\_Line1,  
ExposureActiveMode\_AnyPixels,  
ExposureActiveMode\_AllPixels,

- ```
NUM_EXPOSUREACTIVEMODE }
```
- enum CounterTriggerActivationEnums {  
CounterTriggerActivation\_LevelLow,  
CounterTriggerActivation\_LevelHigh,  
CounterTriggerActivation\_FallingEdge,  
CounterTriggerActivation\_RisingEdge,  
CounterTriggerActivation\_AnyEdge,  
NUM\_COUNTERTRIGGERACTIVATION }
  - enum CounterSelectorEnums {  
CounterSelector\_Counter0,  
CounterSelector\_Counter1,  
NUM\_COUNTERSELECTOR }
  - enum CounterStatusEnums {  
CounterStatus\_CounterIdle,  
CounterStatus\_CounterTriggerWait,  
CounterStatus\_CounterActive,  
CounterStatus\_CounterCompleted,  
CounterStatus\_CounterOverflow,  
NUM\_COUNTERSTATUS }
  - enum CounterTriggerSourceEnums {  
CounterTriggerSource\_Off,  
CounterTriggerSource\_Line0,  
CounterTriggerSource\_Line1,  
CounterTriggerSource\_Line2,  
CounterTriggerSource\_Line3,  
CounterTriggerSource\_UserOutput0,  
CounterTriggerSource\_UserOutput1,  
CounterTriggerSource\_UserOutput2,  
CounterTriggerSource\_UserOutput3,  
CounterTriggerSource\_Counter0Start,  
CounterTriggerSource\_Counter1Start,  
CounterTriggerSource\_Counter0End,  
CounterTriggerSource\_Counter1End,  
CounterTriggerSource\_LogicBlock0,  
CounterTriggerSource\_LogicBlock1,  
CounterTriggerSource\_ExposureStart,  
CounterTriggerSource\_ExposureEnd,  
CounterTriggerSource\_FrameTriggerWait,  
NUM\_COUNTERTRIGGERSOURCE }
  - enum CounterResetSourceEnums {  
CounterResetSource\_Off,  
CounterResetSource\_Line0,  
CounterResetSource\_Line1,  
CounterResetSource\_Line2,  
CounterResetSource\_Line3,  
CounterResetSource\_UserOutput0,  
CounterResetSource\_UserOutput1,  
CounterResetSource\_UserOutput2,  
CounterResetSource\_UserOutput3,  
CounterResetSource\_Counter0Start,  
CounterResetSource\_Counter1Start,  
CounterResetSource\_Counter0End,  
CounterResetSource\_Counter1End,  
CounterResetSource\_LogicBlock0,  
CounterResetSource\_LogicBlock1,  
CounterResetSource\_ExposureStart,  
CounterResetSource\_ExposureEnd,  
CounterResetSource\_FrameTriggerWait,

```

    NUM_COUNTERRESETSOURCE }
• enum CounterEventSourceEnums {
    CounterEventSource_Off,
    CounterEventSource_MHzTick,
    CounterEventSource_Line0,
    CounterEventSource_Line1,
    CounterEventSource_Line2,
    CounterEventSource_Line3,
    CounterEventSource_UserOutput0,
    CounterEventSource_UserOutput1,
    CounterEventSource_UserOutput2,
    CounterEventSource_UserOutput3,
    CounterEventSource_Counter0Start,
    CounterEventSource_Counter1Start,
    CounterEventSource_Counter0End,
    CounterEventSource_Counter1End,
    CounterEventSource_LogicBlock0,
    CounterEventSource_LogicBlock1,
    CounterEventSource_ExposureStart,
    CounterEventSource_ExposureEnd,
    CounterEventSource_FrameTriggerWait,
    NUM_COUNTEREVENTSOURCE }
• enum CounterEventActivationEnums {
    CounterEventActivation_LevelLow,
    CounterEventActivation_LevelHigh,
    CounterEventActivation_FallingEdge,
    CounterEventActivation_RisingEdge,
    CounterEventActivation_AnyEdge,
    NUM_COUNTEREVENTACTIVATION }
• enum CounterResetActivationEnums {
    CounterResetActivation_LevelLow,
    CounterResetActivation_LevelHigh,
    CounterResetActivation_FallingEdge,
    CounterResetActivation_RisingEdge,
    CounterResetActivation_AnyEdge,
    NUM_COUNTERRESETACTIVATION }
• enum DeviceTypeEnums {
    DeviceType_Transmitter,
    DeviceType_Receiver,
    DeviceType_Transceiver,
    DeviceType_Peripheral,
    NUM_DEVICETYPE }
• enum DeviceConnectionStatusEnums {
    DeviceConnectionStatus_Active,
    DeviceConnectionStatus_Inactive,
    NUM_DEVICECONNECTIONSTATUS }
• enum DeviceLinkThroughputLimitModeEnums {
    DeviceLinkThroughputLimitMode_On,
    DeviceLinkThroughputLimitMode_Off,
    NUM_DEVICELINKTHROUGHPUTLIMITMODE }
• enum DeviceLinkHeartbeatModeEnums {
    DeviceLinkHeartbeatMode_On,
    DeviceLinkHeartbeatMode_Off,
    NUM_DEVICELINKHEARTBEATMODE }
• enum DeviceStreamChannelTypeEnums {
    DeviceStreamChannelType_Transmitter,
    DeviceStreamChannelType_Receiver,
    NUM_DEVICESTREAMCHANNELTYPE }

```

- enum DeviceStreamChannelEndiannessEnums {  
DeviceStreamChannelEndianness\_Big,  
DeviceStreamChannelEndianness\_Little,  
NUM\_DEVICESTREAMCHANNELENDIANNESS }
- enum DeviceClockSelectorEnums {  
DeviceClockSelector\_Sensor,  
DeviceClockSelector\_SensorDigitization,  
DeviceClockSelector\_CameraLink,  
NUM\_DEVICECLOCKSELECTOR }
- enum DeviceSerialPortSelectorEnums {  
DeviceSerialPortSelector\_CameraLink,  
NUM\_DEVICSERIALPORTSELECTOR }
- enum DeviceSerialPortBaudRateEnums {  
DeviceSerialPortBaudRate\_Baud9600,  
DeviceSerialPortBaudRate\_Baud19200,  
DeviceSerialPortBaudRate\_Baud38400,  
DeviceSerialPortBaudRate\_Baud57600,  
DeviceSerialPortBaudRate\_Baud115200,  
DeviceSerialPortBaudRate\_Baud230400,  
DeviceSerialPortBaudRate\_Baud460800,  
DeviceSerialPortBaudRate\_Baud921600,  
NUM\_DEVICSERIALPORTBAUDRATE }
- enum SensorTapsEnums {  
SensorTaps\_One,  
SensorTaps\_Two,  
SensorTaps\_Three,  
SensorTaps\_Four,  
SensorTaps\_Eight,  
SensorTaps\_Ten,  
NUM\_SENSORTAPS }
- enum SensorDigitizationTapsEnums {  
SensorDigitizationTaps\_One,  
SensorDigitizationTaps\_Two,  
SensorDigitizationTaps\_Three,  
SensorDigitizationTaps\_Four,  
SensorDigitizationTaps\_Eight,  
SensorDigitizationTaps\_Ten,  
NUM\_SENSORDIGITIZATIONTAPS }
- enum RegionSelectorEnums {  
RegionSelector\_Region0,  
RegionSelector\_Region1,  
RegionSelector\_Region2,  
RegionSelector\_All,  
NUM\_REGIONSELECTOR }
- enum RegionModeEnums {  
RegionMode\_Off,  
RegionMode\_On,  
NUM\_REGIONMODE }
- enum RegionDestinationEnums {  
RegionDestination\_Stream0,  
RegionDestination\_Stream1,  
RegionDestination\_Stream2,  
NUM\_REGIONDESTINATION }
- enum ImageComponentSelectorEnums {

```
ImageComponentSelector_Intensity,  
ImageComponentSelector_Color,  
ImageComponentSelector_Infrared,  
ImageComponentSelector_Ultraviolet,  
ImageComponentSelector_Range,  
ImageComponentSelector_Disparity,  
ImageComponentSelector_Confidence,  
ImageComponentSelector_Scatter,  
NUM_IMAGECOMPONENTSELECTOR }
```

- enum [PixelFormatInfoSelectorEnums](#) {

[PixelFormatInfoSelector\\_Mono1p,](#)  
[PixelFormatInfoSelector\\_Mono2p,](#)  
[PixelFormatInfoSelector\\_Mono4p,](#)  
[PixelFormatInfoSelector\\_Mono8,](#)  
[PixelFormatInfoSelector\\_Mono8s,](#)  
[PixelFormatInfoSelector\\_Mono10,](#)  
[PixelFormatInfoSelector\\_Mono10p,](#)  
[PixelFormatInfoSelector\\_Mono12,](#)  
[PixelFormatInfoSelector\\_Mono12p,](#)  
[PixelFormatInfoSelector\\_Mono14,](#)  
[PixelFormatInfoSelector\\_Mono16,](#)  
[PixelFormatInfoSelector\\_BayerBG8,](#)  
[PixelFormatInfoSelector\\_BayerBG10,](#)  
[PixelFormatInfoSelector\\_BayerBG10p,](#)  
[PixelFormatInfoSelector\\_BayerBG12,](#)  
[PixelFormatInfoSelector\\_BayerBG12p,](#)  
[PixelFormatInfoSelector\\_BayerBG16,](#)  
[PixelFormatInfoSelector\\_BayerGB8,](#)  
[PixelFormatInfoSelector\\_BayerGB10,](#)  
[PixelFormatInfoSelector\\_BayerGB10p,](#)  
[PixelFormatInfoSelector\\_BayerGB12,](#)  
[PixelFormatInfoSelector\\_BayerGB12p,](#)  
[PixelFormatInfoSelector\\_BayerGB16,](#)  
[PixelFormatInfoSelector\\_BayerGR8,](#)  
[PixelFormatInfoSelector\\_BayerGR10,](#)  
[PixelFormatInfoSelector\\_BayerGR10p,](#)  
[PixelFormatInfoSelector\\_BayerGR12,](#)  
[PixelFormatInfoSelector\\_BayerGR12p,](#)  
[PixelFormatInfoSelector\\_BayerGR16,](#)  
[PixelFormatInfoSelector\\_BayerRG8,](#)  
[PixelFormatInfoSelector\\_BayerRG10,](#)  
[PixelFormatInfoSelector\\_BayerRG10p,](#)  
[PixelFormatInfoSelector\\_BayerRG12,](#)  
[PixelFormatInfoSelector\\_BayerRG12p,](#)  
[PixelFormatInfoSelector\\_BayerRG16,](#)  
[PixelFormatInfoSelector\\_RGBa8,](#)  
[PixelFormatInfoSelector\\_RGBa10,](#)  
[PixelFormatInfoSelector\\_RGBa10p,](#)  
[PixelFormatInfoSelector\\_RGBa12,](#)  
[PixelFormatInfoSelector\\_RGBa12p,](#)  
[PixelFormatInfoSelector\\_RGBa14,](#)  
[PixelFormatInfoSelector\\_RGBa16,](#)  
[PixelFormatInfoSelector\\_RGB8,](#)  
[PixelFormatInfoSelector\\_RGB8\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB10,](#)  
[PixelFormatInfoSelector\\_RGB10\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB10p,](#)  
[PixelFormatInfoSelector\\_RGB10p32,](#)  
[PixelFormatInfoSelector\\_RGB12,](#)  
[PixelFormatInfoSelector\\_RGB12\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB12p,](#)  
[PixelFormatInfoSelector\\_RGB14,](#)  
[PixelFormatInfoSelector\\_RGB16,](#)  
[PixelFormatInfoSelector\\_RGB16\\_Planar,](#)  
[PixelFormatInfoSelector\\_RGB565p,](#)  
[PixelFormatInfoSelector\\_BGRa8,](#)  
[PixelFormatInfoSelector\\_BGRa10,](#)  
[PixelFormatInfoSelector\\_BGRa10p,](#)  
[PixelFormatInfoSelector\\_BGRa12,](#)  
[PixelFormatInfoSelector\\_BGRa12p,](#)  
[PixelFormatInfoSelector\\_BGRa14,](#)  
[PixelFormatInfoSelector\\_BGRa16,](#)  
[PixelFormatInfoSelector\\_BGR8,](#)  
[PixelFormatInfoSelector\\_BGR10,](#)  
[PixelFormatInfoSelector\\_BGR10p,](#)

- ```
NUM_PIXELFORMATINFOSELECTOR }
```
- enum DeinterlacingEnums {  
Deinterlacing\_Off,  
Deinterlacing\_LineDuplication,  
Deinterlacing\_Weave,  
NUM\_DEINTERLACING }
  - enum ImageCompressionRateOptionEnums {  
ImageCompressionRateOption\_FixBitrate,  
ImageCompressionRateOption\_FixQuality,  
NUM\_IMAGECOMPRESSIONRATEOPTION }
  - enum ImageCompressionJPEGFormatOptionEnums {  
ImageCompressionJPEGFormatOption\_Lossless,  
ImageCompressionJPEGFormatOption\_BaselineStandard,  
ImageCompressionJPEGFormatOption\_BaselineOptimized,  
ImageCompressionJPEGFormatOption\_Progressive,  
NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION }
  - enum AcquisitionStatusSelectorEnums {  
AcquisitionStatusSelector\_AcquisitionTriggerWait,  
AcquisitionStatusSelector\_AcquisitionActive,  
AcquisitionStatusSelector\_AcquisitionTransfer,  
AcquisitionStatusSelector\_FrameTriggerWait,  
AcquisitionStatusSelector\_FrameActive,  
AcquisitionStatusSelector\_ExposureActive,  
NUM\_ACQUISITIONSTATUSSELECTION }
  - enum ExposureTimeModeEnums {  
ExposureTimeMode\_Common,  
ExposureTimeMode\_Individual,  
NUM\_EXPOSURETIMEMODE }
  - enum ExposureTimeSelectorEnums {  
ExposureTimeSelector\_Common,  
ExposureTimeSelector\_Red,  
ExposureTimeSelector\_Green,  
ExposureTimeSelector\_Blue,  
ExposureTimeSelector\_Cyan,  
ExposureTimeSelector\_Magenta,  
ExposureTimeSelector\_Yellow,  
ExposureTimeSelector\_Infrared,  
ExposureTimeSelector\_Ultraviolet,  
ExposureTimeSelector\_Stage1,  
ExposureTimeSelector\_Stage2,  
NUM\_EXPOSURETIMESELECTION }
  - enum GainAutoBalanceEnums {  
GainAutoBalance\_Off,  
GainAutoBalance\_Once,  
GainAutoBalance\_Continuous,  
NUM\_GAINAUTOBALANCE }
  - enum BlackLevelAutoEnums {  
BlackLevelAuto\_Off,  
BlackLevelAuto\_Once,  
BlackLevelAuto\_Continuous,  
NUM\_BLACKLEVELAUTO }
  - enum BlackLevelAutoBalanceEnums {  
BlackLevelAutoBalance\_Off,  
BlackLevelAutoBalance\_Once,  
BlackLevelAutoBalance\_Continuous,  
NUM\_BLACKLEVELAUTOBALANCE }
  - enum WhiteClipSelectorEnums {



```
WhiteClipSelector_All,  
WhiteClipSelector_Red,  
WhiteClipSelector_Green,  
WhiteClipSelector_Blue,  
WhiteClipSelector_Y,  
WhiteClipSelector_U,  
WhiteClipSelector_V,  
WhiteClipSelector_Tap1,  
WhiteClipSelector_Tap2,  
NUM_WHITECLIPSELECTOR }
```

- enum `TimerSelectorEnums` {  
    `TimerSelector_Timer0`,  
    `TimerSelector_Timer1`,  
    `TimerSelector_Timer2`,  
    `NUM_TIMERSELECTOR` }

- enum `TimerStatusEnums` {  
    `TimerStatus_TimerIdle`,  
    `TimerStatus_TimerTriggerWait`,  
    `TimerStatus_TimerActive`,  
    `TimerStatus_TimerCompleted`,  
    `NUM_TIMERSTATUS` }

- enum `TimerTriggerSourceEnums` {

```

TimerTriggerSource_Off,
TimerTriggerSource_AcquisitionTrigger,
TimerTriggerSource_AcquisitionStart,
TimerTriggerSource_AcquisitionEnd,
TimerTriggerSource_FrameTrigger,
TimerTriggerSource_FrameStart,
TimerTriggerSource_FrameEnd,
TimerTriggerSource_FrameBurstStart,
TimerTriggerSource_FrameBurstEnd,
TimerTriggerSource_LineTrigger,
TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

• enum TimerTriggerActivationEnums {
    TimerTriggerActivation_RisingEdge,
    TimerTriggerActivation_FallingEdge,
    TimerTriggerActivation_AnyEdge,
    TimerTriggerActivation_LevelHigh,
    TimerTriggerActivation_LevelLow,
    NUM_TIMERTRIGGERACTIVATION }

• enum EncoderSelectorEnums {
    EncoderSelector_Encoder0,
    EncoderSelector_Encoder1,
    EncoderSelector_Encoder2,
    NUM_ENCODERSELECTOR }

• enum EncoderSourceAEnums {

```

```
EncoderSourceA_Off,  
EncoderSourceA_Line0,  
EncoderSourceA_Line1,  
EncoderSourceA_Line2,  
NUM_ENCODERSOURCEA }
```

- `enum EncoderSourceBEnums {`  
EncoderSourceB\_Off,  
EncoderSourceB\_Line0,  
EncoderSourceB\_Line1,  
EncoderSourceB\_Line2,  
NUM\_ENCODERSOURCEB }

- `enum EncoderModeEnums {`  
EncoderMode\_FourPhase,  
EncoderMode\_HighResolution,  
NUM\_ENCODERMODE }

- `enum EncoderOutputModeEnums {`  
EncoderOutputMode\_Off,  
EncoderOutputMode\_PositionUp,  
EncoderOutputMode\_PositionDown,  
EncoderOutputMode\_DirectionUp,  
EncoderOutputMode\_DirectionDown,  
EncoderOutputMode\_Motion,  
NUM\_ENCODEROUTPUTMODE }

- `enum EncoderStatusEnums {`  
EncoderStatus\_EncoderUp,  
EncoderStatus\_EncoderDown,  
EncoderStatus\_EncoderIdle,  
EncoderStatus\_EncoderStatic,  
NUM\_ENCODERSTATUS }

- `enum EncoderResetSourceEnums {`

```

EncoderResetSource_Off,
EncoderResetSource_AcquisitionTrigger,
EncoderResetSource_AcquisitionStart,
EncoderResetSource_AcquisitionEnd,
EncoderResetSource_FrameTrigger,
EncoderResetSource_FrameStart,
EncoderResetSource_FrameEnd,
EncoderResetSource_ExposureStart,
EncoderResetSource_ExposureEnd,
EncoderResetSource_Line0,
EncoderResetSource_Line1,
EncoderResetSource_Line2,
EncoderResetSource_Counter0Start,
EncoderResetSource_Counter1Start,
EncoderResetSource_Counter2Start,
EncoderResetSource_Counter0End,
EncoderResetSource_Counter1End,
EncoderResetSource_Counter2End,
EncoderResetSource_Timer0Start,
EncoderResetSource_Timer1Start,
EncoderResetSource_Timer2Start,
EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,
EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }

• enum EncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }

• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }

• enum ActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }

• enum SourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,

```

```

NUM_SOURCESELECTOR }
• enum TransferSelectorEnums {
    TransferSelector_Stream0,
    TransferSelector_Stream1,
    TransferSelector_Stream2,
    TransferSelector_All,
    NUM_TRANSFERSELECTOR }
• enum TransferTriggerSelectorEnums {
    TransferTriggerSelector_TransferStart,
    TransferTriggerSelector_TransferStop,
    TransferTriggerSelector_TransferAbort,
    TransferTriggerSelector_TransferPause,
    TransferTriggerSelector_TransferResume,
    TransferTriggerSelector_TransferActive,
    TransferTriggerSelector_TransferBurstStart,
    TransferTriggerSelector_TransferBurstStop,
    NUM_TRANSFERTRIGGERSELECTOR }
• enum TransferTriggerModeEnums {
    TransferTriggerMode_Off,
    TransferTriggerMode_On,
    NUM_TRANSFERTRIGGERMODE }
• enum TransferTriggerSourceEnums {
    TransferTriggerSource_Line0,
    TransferTriggerSource_Line1,
    TransferTriggerSource_Line2,
    TransferTriggerSource_Counter0Start,
    TransferTriggerSource_Counter1Start,
    TransferTriggerSource_Counter2Start,
    TransferTriggerSource_Counter0End,
    TransferTriggerSource_Counter1End,
    TransferTriggerSource_Counter2End,
    TransferTriggerSource_Timer0Start,
    TransferTriggerSource_Timer1Start,
    TransferTriggerSource_Timer2Start,
    TransferTriggerSource_Timer0End,
    TransferTriggerSource_Timer1End,
    TransferTriggerSource_Timer2End,
    TransferTriggerSource_SoftwareSignal0,
    TransferTriggerSource_SoftwareSignal1,
    TransferTriggerSource_SoftwareSignal2,
    TransferTriggerSource_Action0,
    TransferTriggerSource_Action1,
    TransferTriggerSource_Action2,
    NUM_TRANSFERTRIGGERSOURCE }
• enum TransferTriggerActivationEnums {
    TransferTriggerActivation_RisingEdge,
    TransferTriggerActivation_FallingEdge,
    TransferTriggerActivation_AnyEdge,
    TransferTriggerActivation_LevelHigh,
    TransferTriggerActivation_LevelLow,
    NUM_TRANSFERTRIGGERACTIVATION }
• enum TransferStatusSelectorEnums {
    TransferStatusSelector_Streaming,
    TransferStatusSelector_Paused,
    TransferStatusSelector_Stopping,
    TransferStatusSelector_Stopped,
    TransferStatusSelector_QueueOverflow,
    NUM_TRANSFERSTATUSSELECTOR }

```

- enum [TransferComponentSelectorEnums](#) {  
[TransferComponentSelector\\_Red](#),  
[TransferComponentSelector\\_Green](#),  
[TransferComponentSelector\\_Blue](#),  
[TransferComponentSelector\\_All](#),  
[NUM\\_TRANSFERCOMPONENTSELECTOR](#) }
- enum [Scan3dDistanceUnitEnums](#) {  
[Scan3dDistanceUnit\\_Millimeter](#),  
[Scan3dDistanceUnit\\_Inch](#),  
[NUM\\_SCAN3DDISTANCEUNIT](#) }
- enum [Scan3dCoordinateSystemEnums](#) {  
[Scan3dCoordinateSystem\\_Cartesian](#),  
[Scan3dCoordinateSystem\\_Spherical](#),  
[Scan3dCoordinateSystem\\_Cylindrical](#),  
[NUM\\_SCAN3DCOORDINATESYSTEM](#) }
- enum [Scan3dOutputModeEnums](#) {  
[Scan3dOutputMode\\_UncalibratedC](#),  
[Scan3dOutputMode\\_CalibratedABC\\_Grid](#),  
[Scan3dOutputMode\\_CalibratedABC\\_PointCloud](#),  
[Scan3dOutputMode\\_CalibratedAC](#),  
[Scan3dOutputMode\\_CalibratedAC\\_Linescan](#),  
[Scan3dOutputMode\\_CalibratedC](#),  
[Scan3dOutputMode\\_CalibratedC\\_Linescan](#),  
[Scan3dOutputMode\\_RectifiedC](#),  
[Scan3dOutputMode\\_RectifiedC\\_Linescan](#),  
[Scan3dOutputMode\\_DisparityC](#),  
[Scan3dOutputMode\\_DisparityC\\_Linescan](#),  
[NUM\\_SCAN3DOUTPUTMODE](#) }
- enum [Scan3dCoordinateSystemReferenceEnums](#) {  
[Scan3dCoordinateSystemReference\\_Anchor](#),  
[Scan3dCoordinateSystemReference\\_Transformed](#),  
[NUM\\_SCAN3DCOORDINATESYSTEMREFERENCE](#) }
- enum [Scan3dCoordinateSelectorEnums](#) {  
[Scan3dCoordinateSelector\\_CoordinateA](#),  
[Scan3dCoordinateSelector\\_CoordinateB](#),  
[Scan3dCoordinateSelector\\_CoordinateC](#),  
[NUM\\_SCAN3DCOORDINATESELECTOR](#) }
- enum [Scan3dCoordinateTransformSelectorEnums](#) {  
[Scan3dCoordinateTransformSelector\\_RotationX](#),  
[Scan3dCoordinateTransformSelector\\_RotationY](#),  
[Scan3dCoordinateTransformSelector\\_RotationZ](#),  
[Scan3dCoordinateTransformSelector\\_TranslationX](#),  
[Scan3dCoordinateTransformSelector\\_TranslationY](#),  
[Scan3dCoordinateTransformSelector\\_TranslationZ](#),  
[NUM\\_SCAN3DCOORDINATETRANSFORMSELECTOR](#) }
- enum [Scan3dCoordinateReferenceSelectorEnums](#) {  
[Scan3dCoordinateReferenceSelector\\_RotationX](#),  
[Scan3dCoordinateReferenceSelector\\_RotationY](#),  
[Scan3dCoordinateReferenceSelector\\_RotationZ](#),  
[Scan3dCoordinateReferenceSelector\\_TranslationX](#),  
[Scan3dCoordinateReferenceSelector\\_TranslationY](#),  
[Scan3dCoordinateReferenceSelector\\_TranslationZ](#),  
[NUM\\_SCAN3DCOORDINATEREFERENCESELECTOR](#) }
- enum [ChunkImageComponentEnums](#) {

```
ChunkImageComponent_Intensity,  
ChunkImageComponent_Color,  
ChunkImageComponent_Infrared,  
ChunkImageComponent_Ultraviolet,  
ChunkImageComponent_Range,  
ChunkImageComponent_Disparity,  
ChunkImageComponent_Confidence,  
ChunkImageComponent_Scatter,  
NUM_CHUNKIMAGECOMPONENT }  
• enum ChunkCounterSelectorEnums {  
    ChunkCounterSelector_Counter0,  
    ChunkCounterSelector_Counter1,  
    ChunkCounterSelector_Counter2,  
    NUM_CHUNKCOUNTERSELECTOR }  
• enum ChunkTimerSelectorEnums {  
    ChunkTimerSelector_Timer0,  
    ChunkTimerSelector_Timer1,  
    ChunkTimerSelector_Timer2,  
    NUM_CHUNKTIMERSELECTOR }  
• enum ChunkEncoderSelectorEnums {  
    ChunkEncoderSelector_Encoder0,  
    ChunkEncoderSelector_Encoder1,  
    ChunkEncoderSelector_Encoder2,  
    NUM_CHUNKENCODERSELECTOR }  
• enum ChunkEncoderStatusEnums {  
    ChunkEncoderStatus_EncoderUp,  
    ChunkEncoderStatus_EncoderDown,  
    ChunkEncoderStatus_EncoderIdle,  
    ChunkEncoderStatus_EncoderStatic,  
    NUM_CHUNKENCODERSTATUS }  
• enum ChunkExposureTimeSelectorEnums {  
    ChunkExposureTimeSelector_Common,  
    ChunkExposureTimeSelector_Red,  
    ChunkExposureTimeSelector_Green,  
    ChunkExposureTimeSelector_Blue,  
    ChunkExposureTimeSelector_Cyan,  
    ChunkExposureTimeSelector_Magenta,  
    ChunkExposureTimeSelector_Yellow,  
    ChunkExposureTimeSelector_Infrared,  
    ChunkExposureTimeSelector_Ultraviolet,  
    ChunkExposureTimeSelector_Stage1,  
    ChunkExposureTimeSelector_Stage2,  
    NUM_CHUNKEXPOSURETIMESELECTOR }  
• enum ChunkSourceIDEnums {  
    ChunkSourceID_Source0,  
    ChunkSourceID_Source1,  
    ChunkSourceID_Source2,  
    NUM_CHUNKSOURCEID }  
• enum ChunkRegionIDEnums {  
    ChunkRegionID_Region0,  
    ChunkRegionID_Region1,  
    ChunkRegionID_Region2,  
    NUM_CHUNKREGIONID }  
• enum ChunkTransferStreamIDEnums {  
    ChunkTransferStreamID_Stream0,  
    ChunkTransferStreamID_Stream1,  
    ChunkTransferStreamID_Stream2,  
    ChunkTransferStreamID_Stream3,
```

```
NUM_CHUNKTRANSFERSTREAMID }
```

- enum ChunkScan3dDistanceUnitEnums {  
 ChunkScan3dDistanceUnit\_Millimeter,  
 ChunkScan3dDistanceUnit\_Inch,  
 NUM\_CHUNKSCAN3DDISTANCEUNIT }
- enum ChunkScan3dOutputModeEnums {  
 ChunkScan3dOutputMode\_UncalibratedC,  
 ChunkScan3dOutputMode\_CalibratedABC\_Grid,  
 ChunkScan3dOutputMode\_CalibratedABC\_PointCloud,  
 ChunkScan3dOutputMode\_CalibratedAC,  
 ChunkScan3dOutputMode\_CalibratedAC\_Linescan,  
 ChunkScan3dOutputMode\_CalibratedC,  
 ChunkScan3dOutputMode\_CalibratedC\_Linescan,  
 ChunkScan3dOutputMode\_RectifiedC,  
 ChunkScan3dOutputMode\_RectifiedC\_Linescan,  
 ChunkScan3dOutputMode\_DisparityC,  
 ChunkScan3dOutputMode\_DisparityC\_Linescan,  
 NUM\_CHUNKSCAN3DOUTPUTMODE }
- enum ChunkScan3dCoordinateSystemEnums {  
 ChunkScan3dCoordinateSystem\_Cartesian,  
 ChunkScan3dCoordinateSystem\_Spherical,  
 ChunkScan3dCoordinateSystem\_Cylindrical,  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEM }
- enum ChunkScan3dCoordinateSystemReferenceEnums {  
 ChunkScan3dCoordinateSystemReference\_Anchor,  
 ChunkScan3dCoordinateSystemReference\_Transformed,  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }
- enum ChunkScan3dCoordinateSelectorEnums {  
 ChunkScan3dCoordinateSelector\_CoordinateA,  
 ChunkScan3dCoordinateSelector\_CoordinateB,  
 ChunkScan3dCoordinateSelector\_CoordinateC,  
 NUM\_CHUNKSCAN3DCOORDINATESELECTOR }
- enum ChunkScan3dCoordinateTransformSelectorEnums {  
 ChunkScan3dCoordinateTransformSelector\_RotationX,  
 ChunkScan3dCoordinateTransformSelector\_RotationY,  
 ChunkScan3dCoordinateTransformSelector\_RotationZ,  
 ChunkScan3dCoordinateTransformSelector\_TranslationX,  
 ChunkScan3dCoordinateTransformSelector\_TranslationY,  
 ChunkScan3dCoordinateTransformSelector\_TranslationZ,  
 NUM\_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }
- enum ChunkScan3dCoordinateReferenceSelectorEnums {  
 ChunkScan3dCoordinateReferenceSelector\_RotationX,  
 ChunkScan3dCoordinateReferenceSelector\_RotationY,  
 ChunkScan3dCoordinateReferenceSelector\_RotationZ,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationX,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationY,  
 ChunkScan3dCoordinateReferenceSelector\_TranslationZ,  
 NUM\_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }
- enum DeviceTapGeometryEnums {



```

DeviceTapGeometry_Geometry_1X_1Y,
DeviceTapGeometry_Geometry_1X2_1Y,
DeviceTapGeometry_Geometry_1X2_1Y2,
DeviceTapGeometry_Geometry_2X_1Y,
DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
DeviceTapGeometry_Geometry_2XE_1Y2,
DeviceTapGeometry_Geometry_2XM_1Y,
DeviceTapGeometry_Geometry_2XM_1Y2,
DeviceTapGeometry_Geometry_1X_1Y2,
DeviceTapGeometry_Geometry_1X_2YE,
DeviceTapGeometry_Geometry_1X3_1Y,
DeviceTapGeometry_Geometry_3X_1Y,
DeviceTapGeometry_Geometry_1X,
DeviceTapGeometry_Geometry_1X2,
DeviceTapGeometry_Geometry_2X,
DeviceTapGeometry_Geometry_2XE,
DeviceTapGeometry_Geometry_2XM,
DeviceTapGeometry_Geometry_1X3,
DeviceTapGeometry_Geometry_3X,
DeviceTapGeometry_Geometry_1X4_1Y,
DeviceTapGeometry_Geometry_4X_1Y,
DeviceTapGeometry_Geometry_2X2_1Y,
DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
DeviceTapGeometry_Geometry_1X2_2YE,
DeviceTapGeometry_Geometry_2X_2YE,
DeviceTapGeometry_Geometry_2XE_2YE,
DeviceTapGeometry_Geometry_2XM_2YE,
DeviceTapGeometry_Geometry_1X4,
DeviceTapGeometry_Geometry_4X,
DeviceTapGeometry_Geometry_2X2,
DeviceTapGeometry_Geometry_2X2E,
DeviceTapGeometry_Geometry_2X2M,
DeviceTapGeometry_Geometry_1X8_1Y,
DeviceTapGeometry_Geometry_8X_1Y,
DeviceTapGeometry_Geometry_4X2_1Y,
DeviceTapGeometry_Geometry_2X2E_2YE,
DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

• enum GevPhysicalLinkConfigurationEnums {
    GevPhysicalLinkConfiguration_SingleLink,
    GevPhysicalLinkConfiguration_MultiLink,
    GevPhysicalLinkConfiguration_StaticLAG,
    GevPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVPHYSICALLINKCONFIGURATION }

• enum GevCurrentPhysicalLinkConfigurationEnums {
    GevCurrentPhysicalLinkConfiguration_SingleLink,
    GevCurrentPhysicalLinkConfiguration_MultiLink,
    GevCurrentPhysicalLinkConfiguration_StaticLAG,
    GevCurrentPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }

```

- enum [GevIPConfigurationStatusEnums](#) {  
    [GevIPConfigurationStatus\\_None](#),  
    [GevIPConfigurationStatus\\_PersistentIP](#),  
    [GevIPConfigurationStatus\\_DHCP](#),  
    [GevIPConfigurationStatus\\_LLA](#),  
    [GevIPConfigurationStatus\\_ForceIP](#),  
    [NUM\\_GEVIPCONFIGURATIONSTATUS](#) }
  
- enum [GevGVCPExtendedStatusCodesSelectorEnums](#) {  
    [GevGVCPExtendedStatusCodesSelector\\_Version1\\_1](#),  
    [GevGVCPExtendedStatusCodesSelector\\_Version2\\_0](#),  
    [NUM\\_GEVGVCPEXTENDEDSTATUSCODESSELECTOR](#) }
  
- enum [GevGVSPExtendedIDModeEnums](#) {  
    [GevGVSPExtendedIDMode\\_Off](#),  
    [GevGVSPExtendedIDMode\\_On](#),  
    [NUM\\_GEVGVSPEXTENDEDIDMODE](#) }
  
- enum [CIConfigurationEnums](#) {  
    [CIConfiguration\\_Base](#),  
    [CIConfiguration\\_Medium](#),  
    [CIConfiguration\\_Full](#),  
    [CIConfiguration\\_DualBase](#),  
    [CIConfiguration\\_EightyBit](#),  
    [NUM\\_CLCONFIGURATION](#) }
  
- enum [CITimeSlotsCountEnums](#) {  
    [CITimeSlotsCount\\_One](#),  
    [CITimeSlotsCount\\_Two](#),  
    [CITimeSlotsCount\\_Three](#),  
    [NUM\\_CLTIMESLOTSCOUNT](#) }
  
- enum [CxpLinkConfigurationStatusEnums](#) {

```
CxpLinkConfigurationStatus_None,  
CxpLinkConfigurationStatus_Pending,  
CxpLinkConfigurationStatus_CXP1_X1,  
CxpLinkConfigurationStatus_CXP2_X1,  
CxpLinkConfigurationStatus_CXP3_X1,  
CxpLinkConfigurationStatus_CXP5_X1,  
CxpLinkConfigurationStatus_CXP6_X1,  
CxpLinkConfigurationStatus_CXP1_X2,  
CxpLinkConfigurationStatus_CXP2_X2,  
CxpLinkConfigurationStatus_CXP3_X2,  
CxpLinkConfigurationStatus_CXP5_X2,  
CxpLinkConfigurationStatus_CXP6_X2,  
CxpLinkConfigurationStatus_CXP1_X3,  
CxpLinkConfigurationStatus_CXP2_X3,  
CxpLinkConfigurationStatus_CXP3_X3,  
CxpLinkConfigurationStatus_CXP5_X3,  
CxpLinkConfigurationStatus_CXP6_X3,  
CxpLinkConfigurationStatus_CXP1_X4,  
CxpLinkConfigurationStatus_CXP2_X4,  
CxpLinkConfigurationStatus_CXP3_X4,  
CxpLinkConfigurationStatus_CXP5_X4,  
CxpLinkConfigurationStatus_CXP6_X4,  
CxpLinkConfigurationStatus_CXP1_X5,  
CxpLinkConfigurationStatus_CXP2_X5,  
CxpLinkConfigurationStatus_CXP3_X5,  
CxpLinkConfigurationStatus_CXP5_X5,  
CxpLinkConfigurationStatus_CXP6_X5,  
CxpLinkConfigurationStatus_CXP1_X6,  
CxpLinkConfigurationStatus_CXP2_X6,  
CxpLinkConfigurationStatus_CXP3_X6,  
CxpLinkConfigurationStatus_CXP5_X6,  
CxpLinkConfigurationStatus_CXP6_X6,  
NUM_CXPLINKCONFIGURATIONSTATUS }
```

• enum [CxpLinkConfigurationPreferredEnums](#) {

```
CxpLinkConfigurationPreferred_CXP1_X1,  
CxpLinkConfigurationPreferred_CXP2_X1,  
CxpLinkConfigurationPreferred_CXP3_X1,  
CxpLinkConfigurationPreferred_CXP5_X1,  
CxpLinkConfigurationPreferred_CXP6_X1,  
CxpLinkConfigurationPreferred_CXP1_X2,  
CxpLinkConfigurationPreferred_CXP2_X2,  
CxpLinkConfigurationPreferred_CXP3_X2,  
CxpLinkConfigurationPreferred_CXP5_X2,  
CxpLinkConfigurationPreferred_CXP6_X2,  
CxpLinkConfigurationPreferred_CXP1_X3,  
CxpLinkConfigurationPreferred_CXP2_X3,  
CxpLinkConfigurationPreferred_CXP3_X3,  
CxpLinkConfigurationPreferred_CXP5_X3,  
CxpLinkConfigurationPreferred_CXP6_X3,  
CxpLinkConfigurationPreferred_CXP1_X4,  
CxpLinkConfigurationPreferred_CXP2_X4,  
CxpLinkConfigurationPreferred_CXP3_X4,  
CxpLinkConfigurationPreferred_CXP5_X4,  
CxpLinkConfigurationPreferred_CXP6_X4,  
CxpLinkConfigurationPreferred_CXP1_X5,  
CxpLinkConfigurationPreferred_CXP2_X5,  
CxpLinkConfigurationPreferred_CXP3_X5,  
CxpLinkConfigurationPreferred_CXP5_X5,  
CxpLinkConfigurationPreferred_CXP6_X5,  
CxpLinkConfigurationPreferred_CXP1_X6,  
CxpLinkConfigurationPreferred_CXP2_X6,  
CxpLinkConfigurationPreferred_CXP3_X6,  
CxpLinkConfigurationPreferred_CXP5_X6,  
CxpLinkConfigurationPreferred_CXP6_X6,  
NUM_CXPLINKCONFIGURATIONPREFERRED }
```

• enum [CxpLinkConfigurationEnums](#) {

```

CxpLinkConfiguration_Auto,
CxpLinkConfiguration_CXP1_X1,
CxpLinkConfiguration_CXP2_X1,
CxpLinkConfiguration_CXP3_X1,
CxpLinkConfiguration_CXP5_X1,
CxpLinkConfiguration_CXP6_X1,
CxpLinkConfiguration_CXP1_X2,
CxpLinkConfiguration_CXP2_X2,
CxpLinkConfiguration_CXP3_X2,
CxpLinkConfiguration_CXP5_X2,
CxpLinkConfiguration_CXP6_X2,
CxpLinkConfiguration_CXP1_X3,
CxpLinkConfiguration_CXP2_X3,
CxpLinkConfiguration_CXP3_X3,
CxpLinkConfiguration_CXP5_X3,
CxpLinkConfiguration_CXP6_X3,
CxpLinkConfiguration_CXP1_X4,
CxpLinkConfiguration_CXP2_X4,
CxpLinkConfiguration_CXP3_X4,
CxpLinkConfiguration_CXP5_X4,
CxpLinkConfiguration_CXP6_X4,
CxpLinkConfiguration_CXP1_X5,
CxpLinkConfiguration_CXP2_X5,
CxpLinkConfiguration_CXP3_X5,
CxpLinkConfiguration_CXP5_X5,
CxpLinkConfiguration_CXP6_X5,
CxpLinkConfiguration_CXP1_X6,
CxpLinkConfiguration_CXP2_X6,
CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }

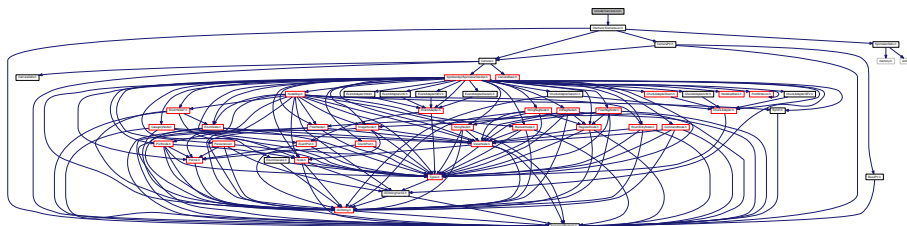
• enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }

• enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }

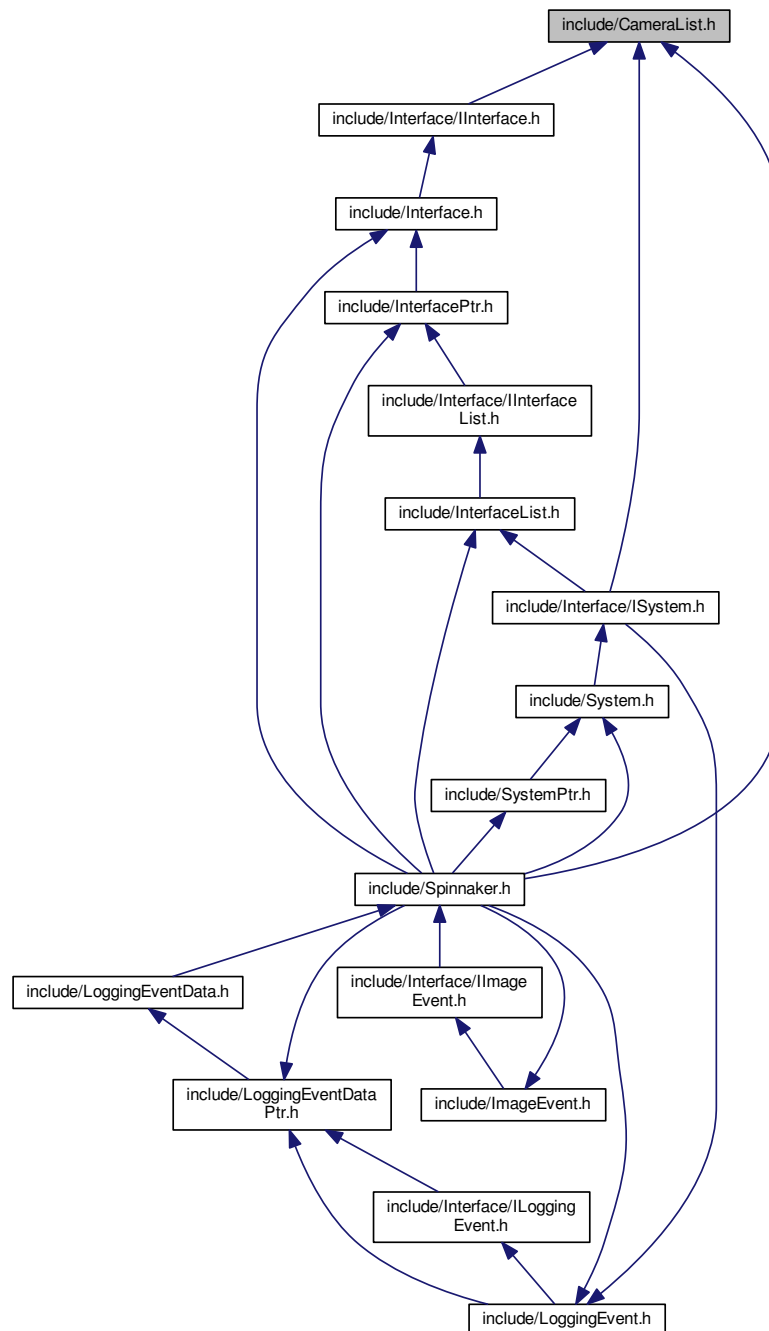
```

## 11.9 include/CameraList.h File Reference

Include dependency graph for CameraList.h:



This graph shows which files directly or indirectly include this file:



## Classes

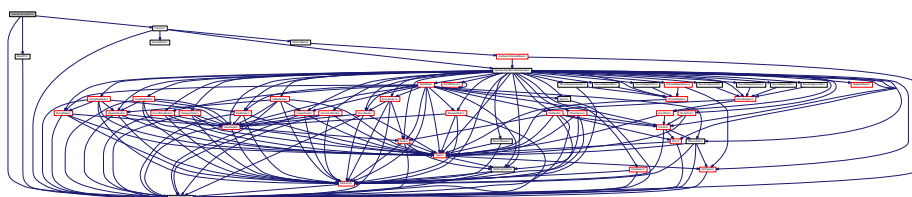
- class [CameraList](#)  
*Used to hold a list of camera objects.*

## Namespaces

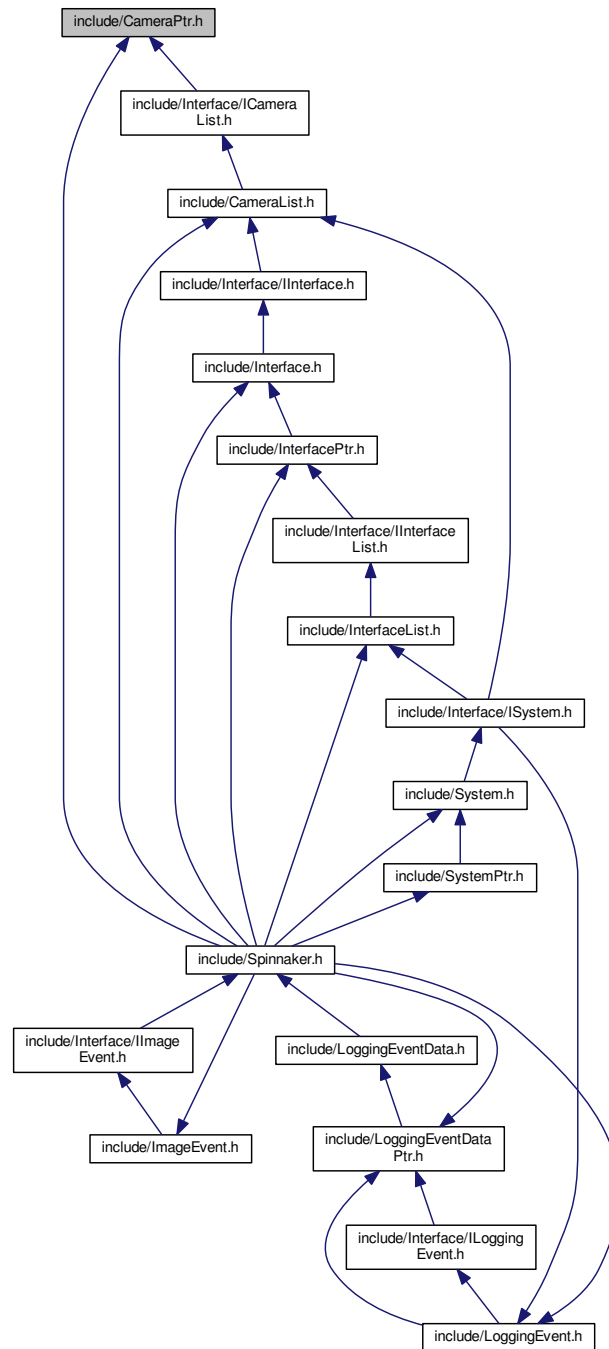
- [Spinnaker](#)

## 11.10 include/CameraPtr.h File Reference

Include dependency graph for CameraPtr.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [CameraPtr](#)  
A reference tracked pointer to a camera object.

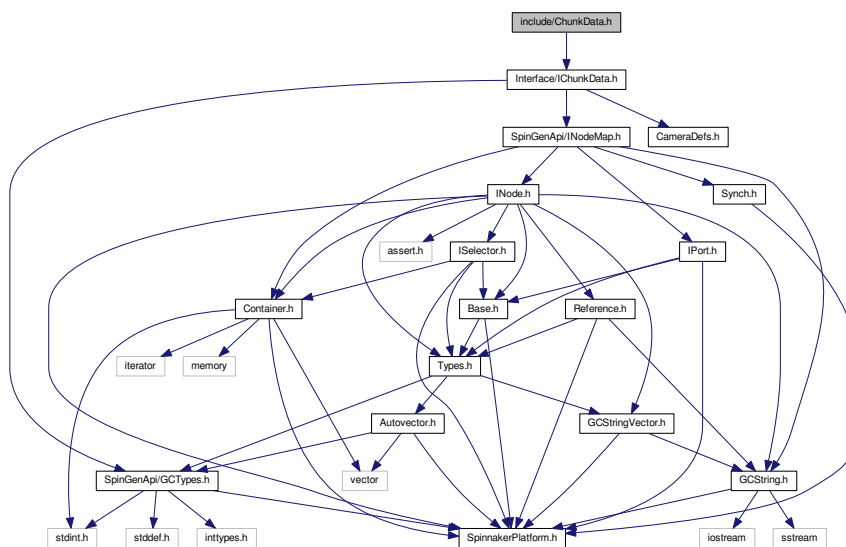
## Namespaces

- [Spinnaker](#)

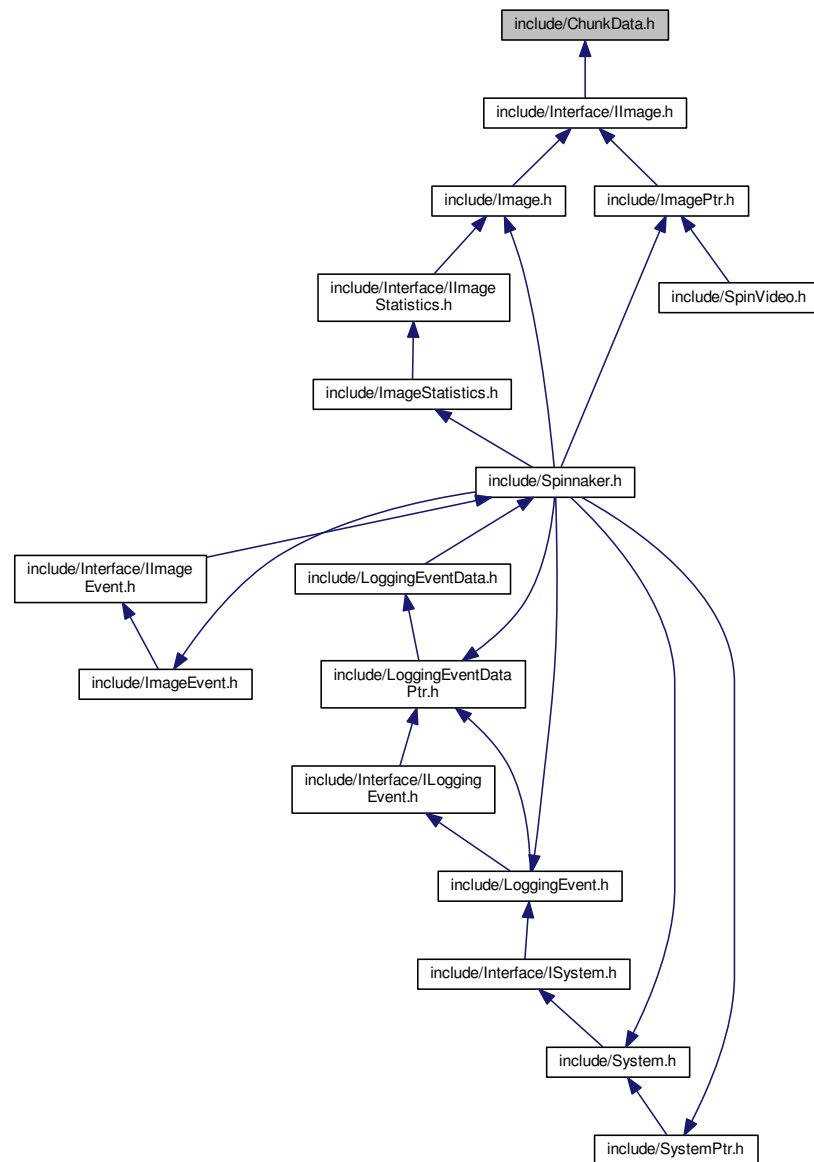


## 11.11 include/ChunkData.h File Reference

Include dependency graph for ChunkData.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ChunkData](#)

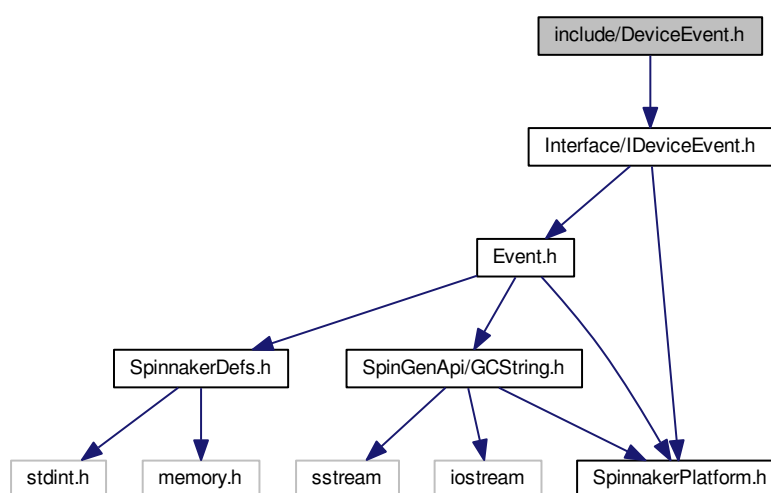
*The chunk data which contains additional information about an image.*

## Namespaces

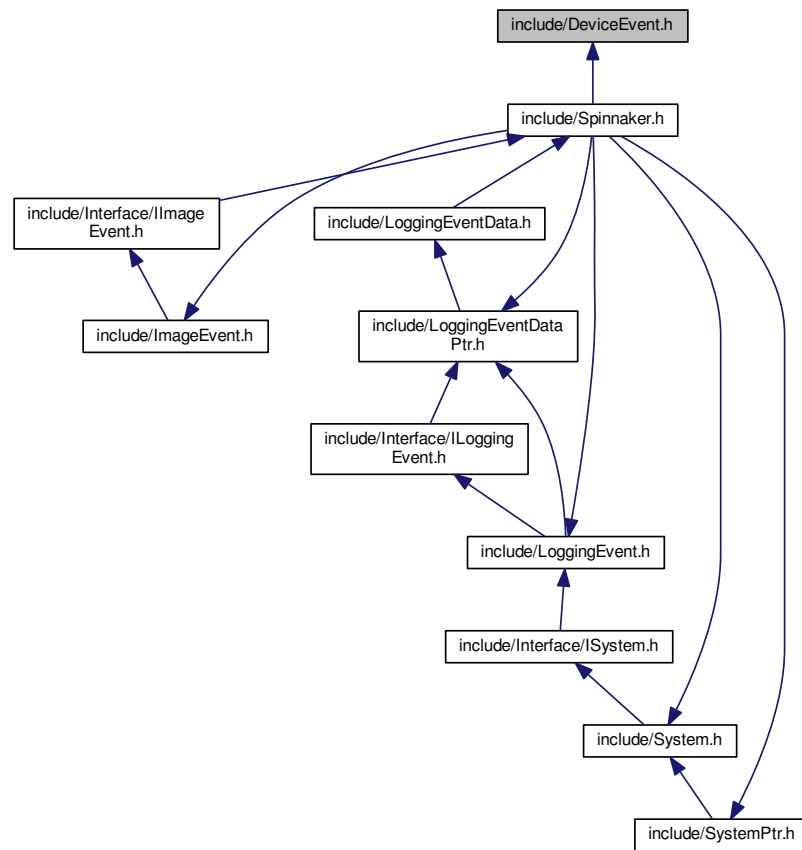
- [Spinnaker](#)

## 11.12 include/DeviceEvent.h File Reference

Include dependency graph for DeviceEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [DeviceEvent](#)

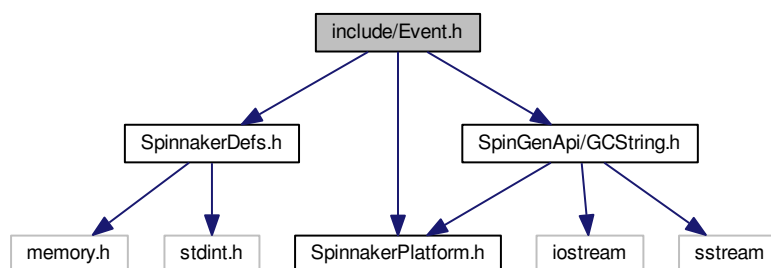
*A handler to device events.*

## Namespaces

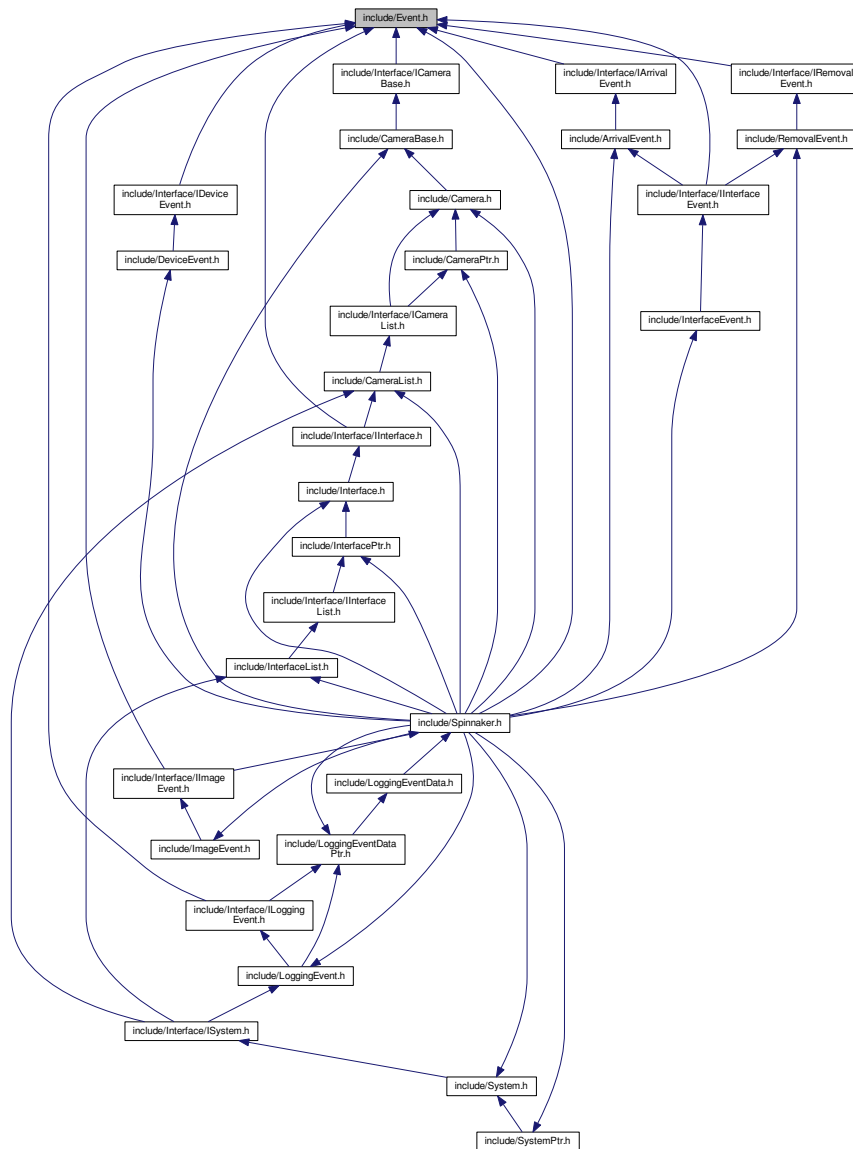
- [Spinnaker](#)

## 11.13 include/Event.h File Reference

Include dependency graph for Event.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Event](#)

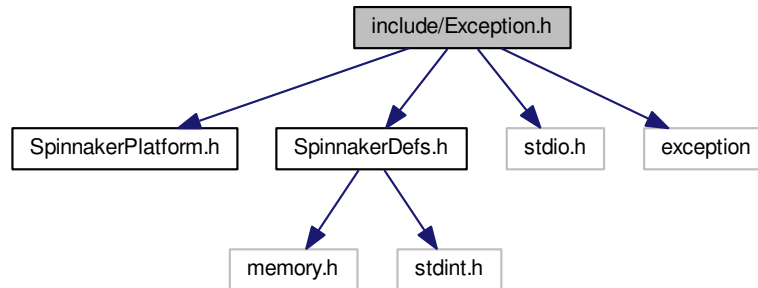
*The base class for all event types.*

## Namespaces

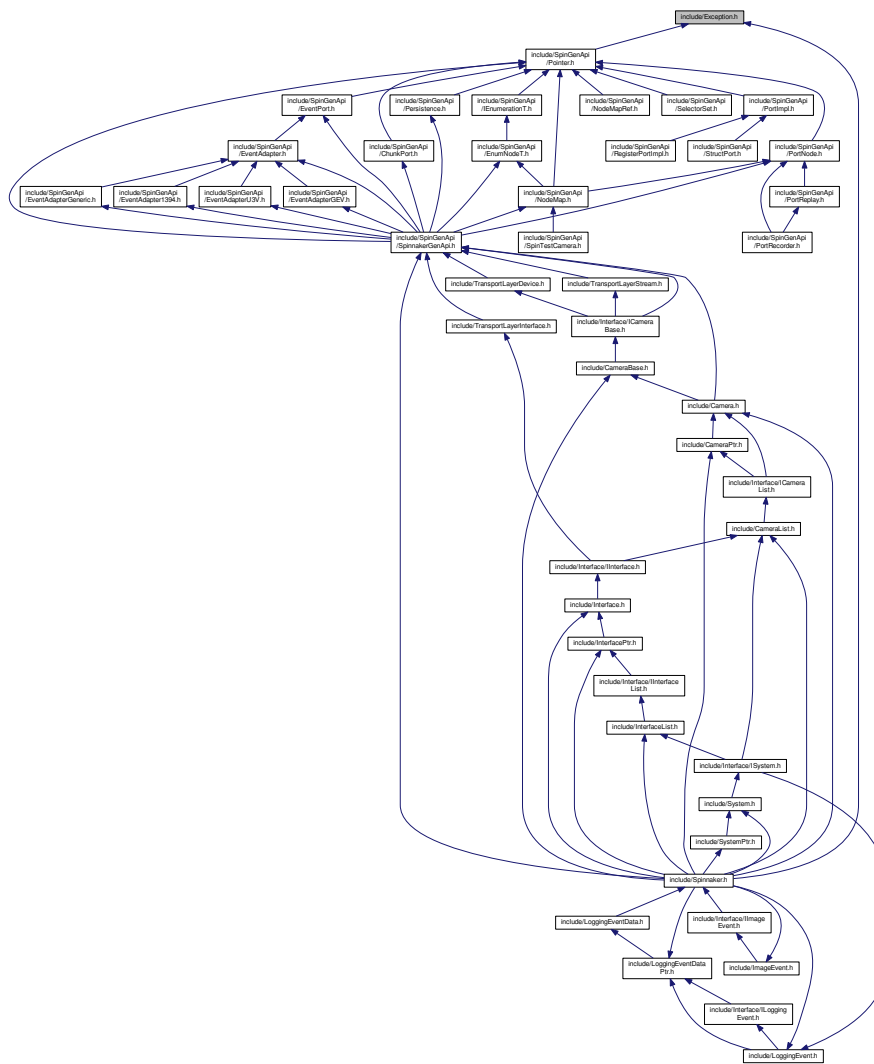
- [Spinnaker](#)

## 11.14 include/Exception.h File Reference

Include dependency graph for Exception.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Exception](#)

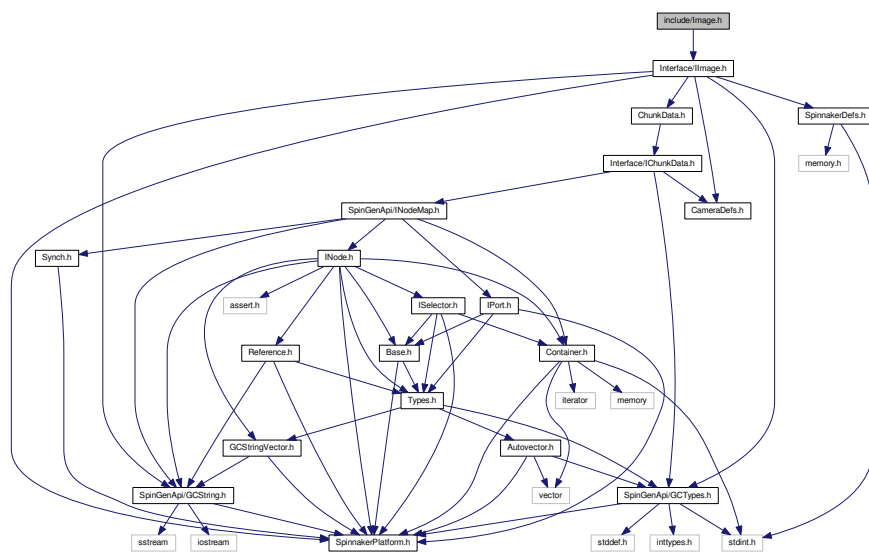
The [Exception](#) object represents an error that is returned from the library.

## Namespaces

- [Spinnaker](#)

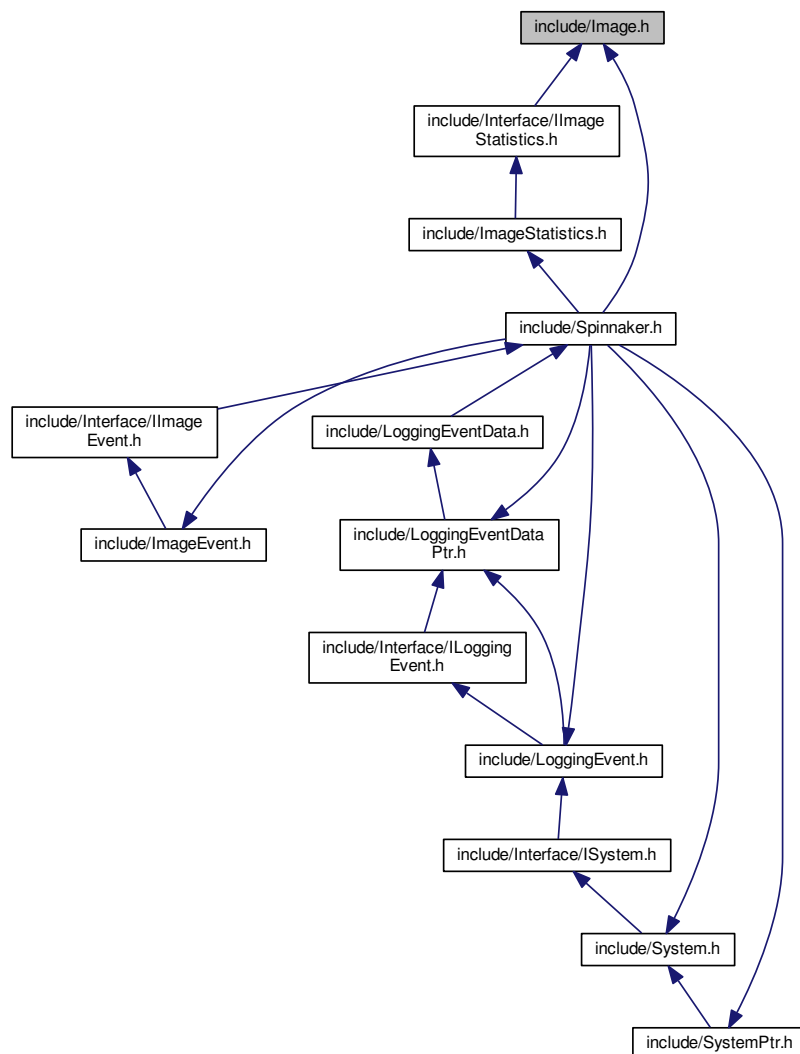
## 11.15 include/Image.h File Reference

Include dependency graph for Image.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [Image](#)

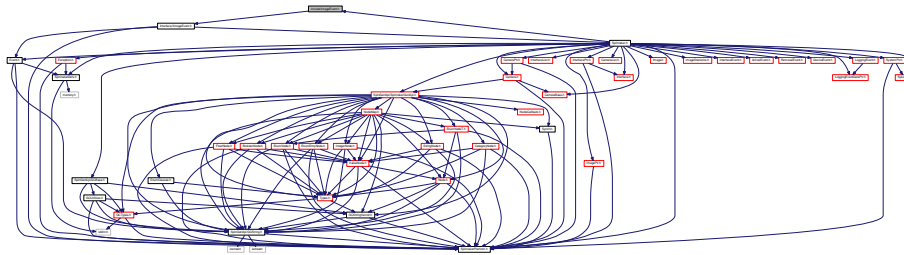
*The image object class.*

## Namespaces

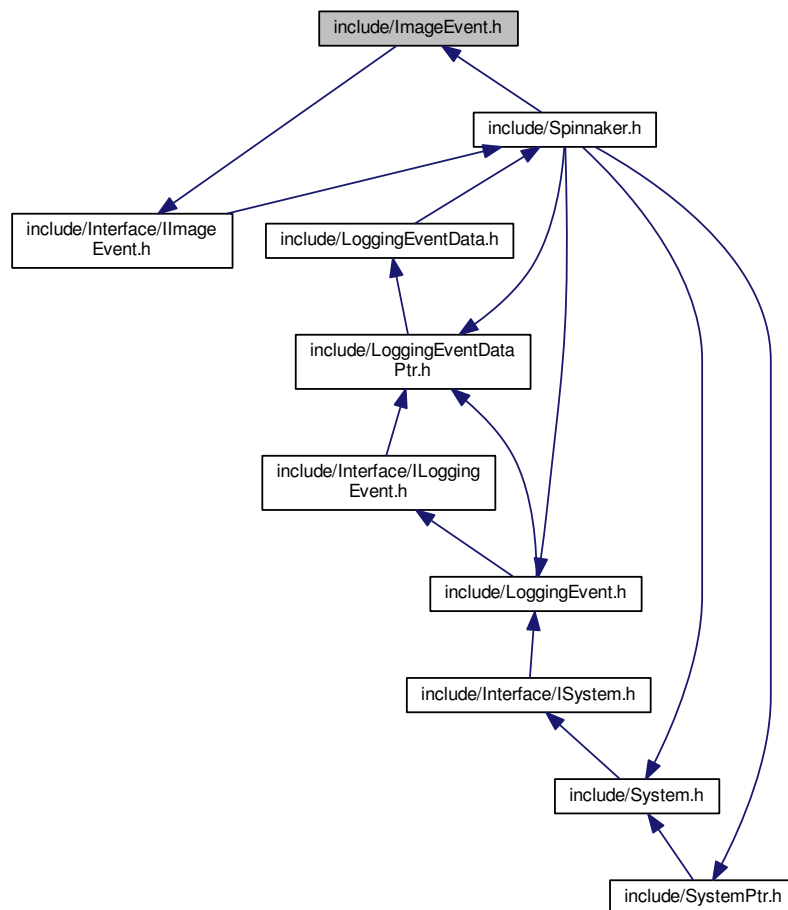
- [Spinnaker](#)

## 11.16 include/ImageEvent.h File Reference

Include dependency graph for ImageEvent.h:



This graph shows which files directly or indirectly include this file:

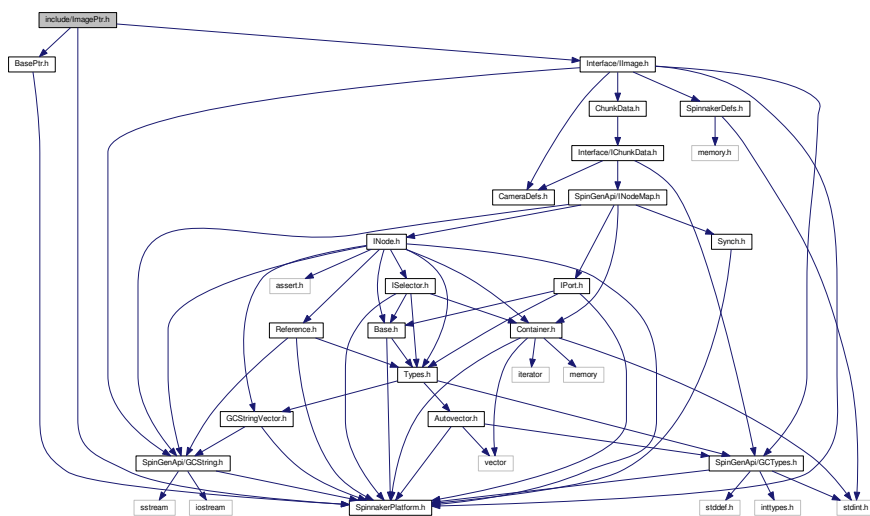


### Classes

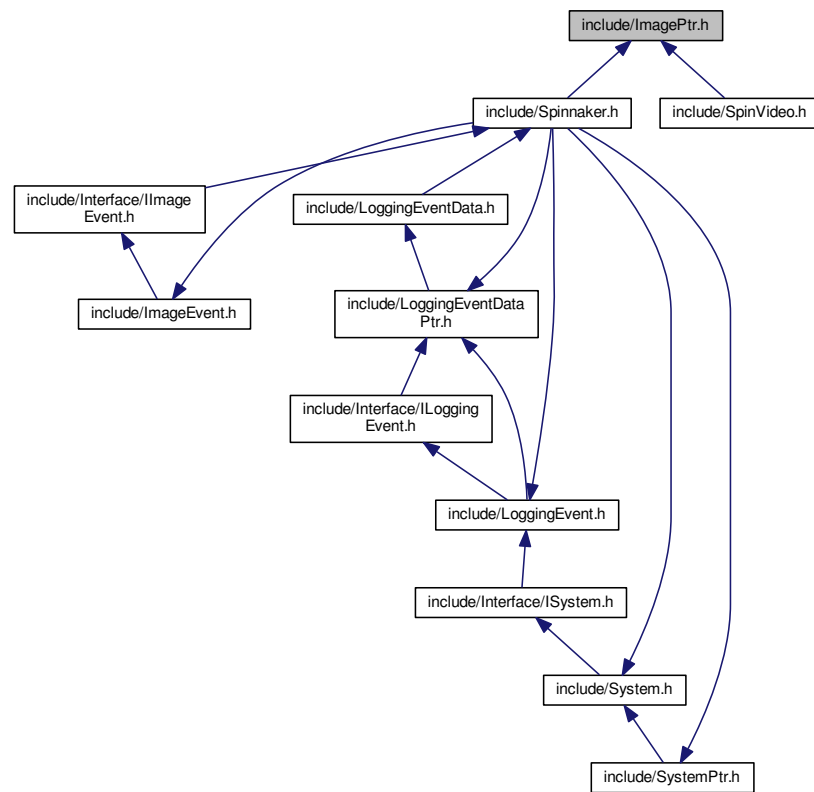
- class [ImageEvent](#)  
A handler for capturing image arrival events.

- Spinnaker

Include dependency graph for ImagePtr.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ImagePtr](#)

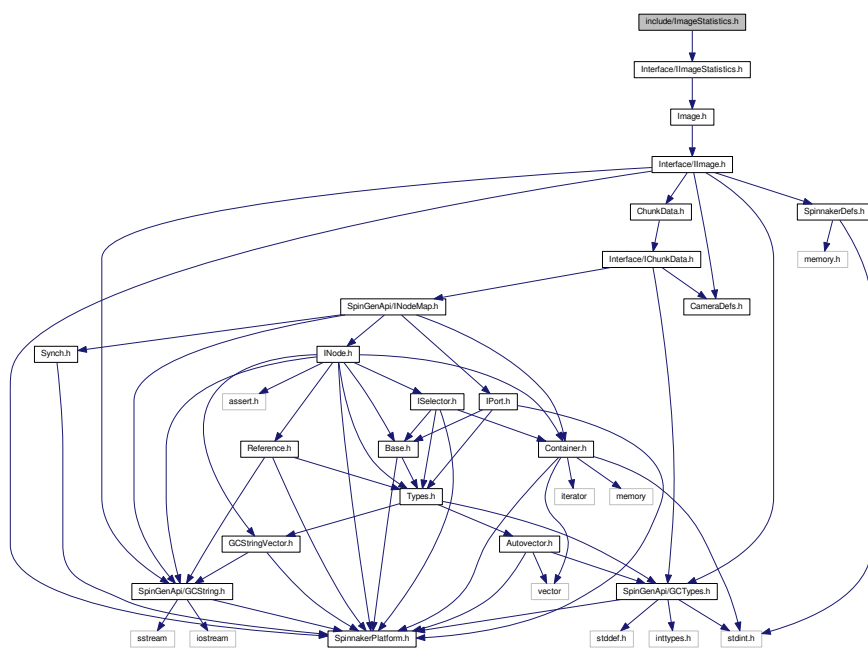
*A reference tracked pointer to an image object.*

## Namespaces

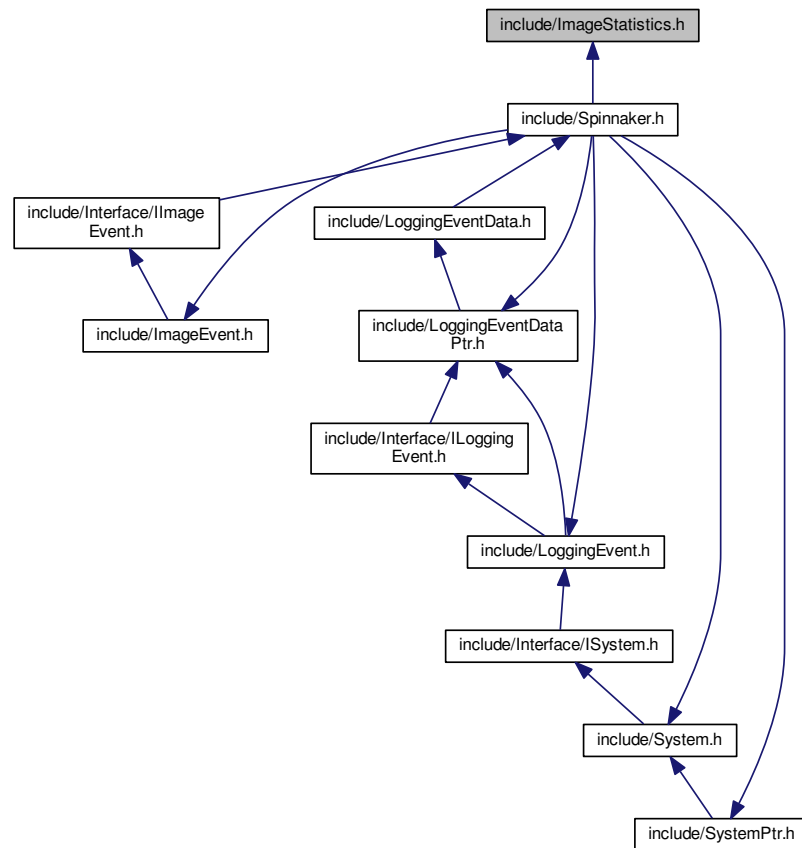
- [Spinnaker](#)

## 11.18 include/ImageStatistics.h File Reference

Include dependency graph for ImageStatistics.h:



This graph shows which files directly or indirectly include this file:



## Classes

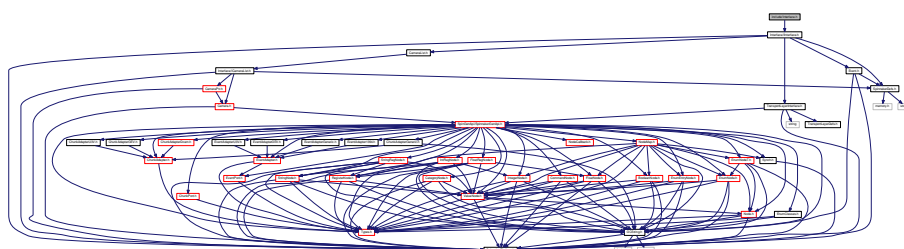
- class [ImageStatistics](#)  
*Represents image statistics for an image.*

## Namespaces

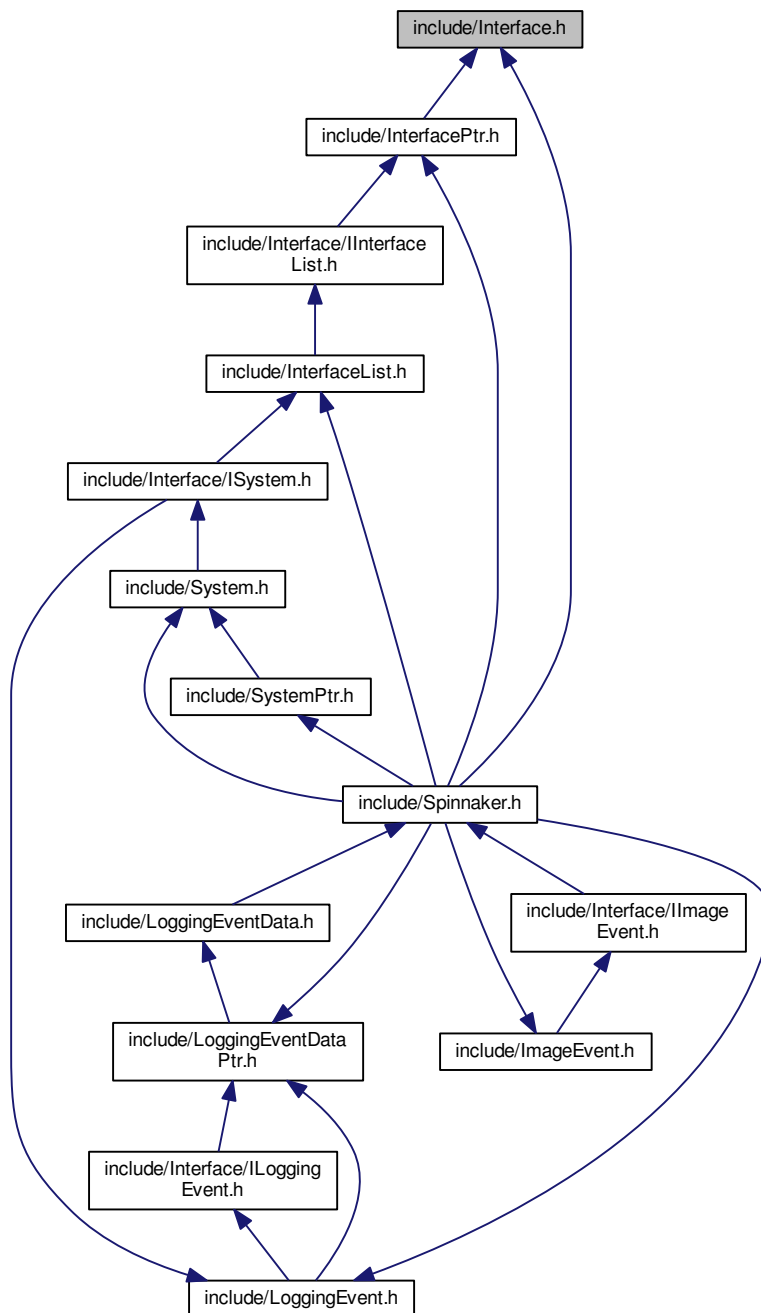
- [Spinnaker](#)

## 11.19 include/Interface.h File Reference

Include dependency graph for `Interface.h`:



This graph shows which files directly or indirectly include this file:



## Classes

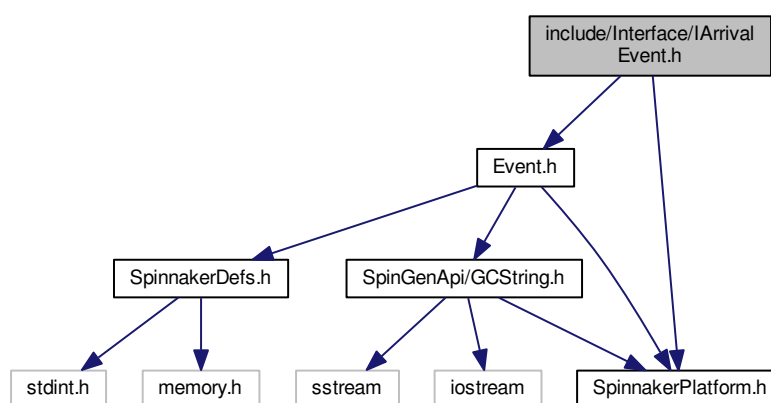
- class [Interface](#)  
An interface object which holds a list of cameras.

## Namespaces

- [Spinnaker](#)

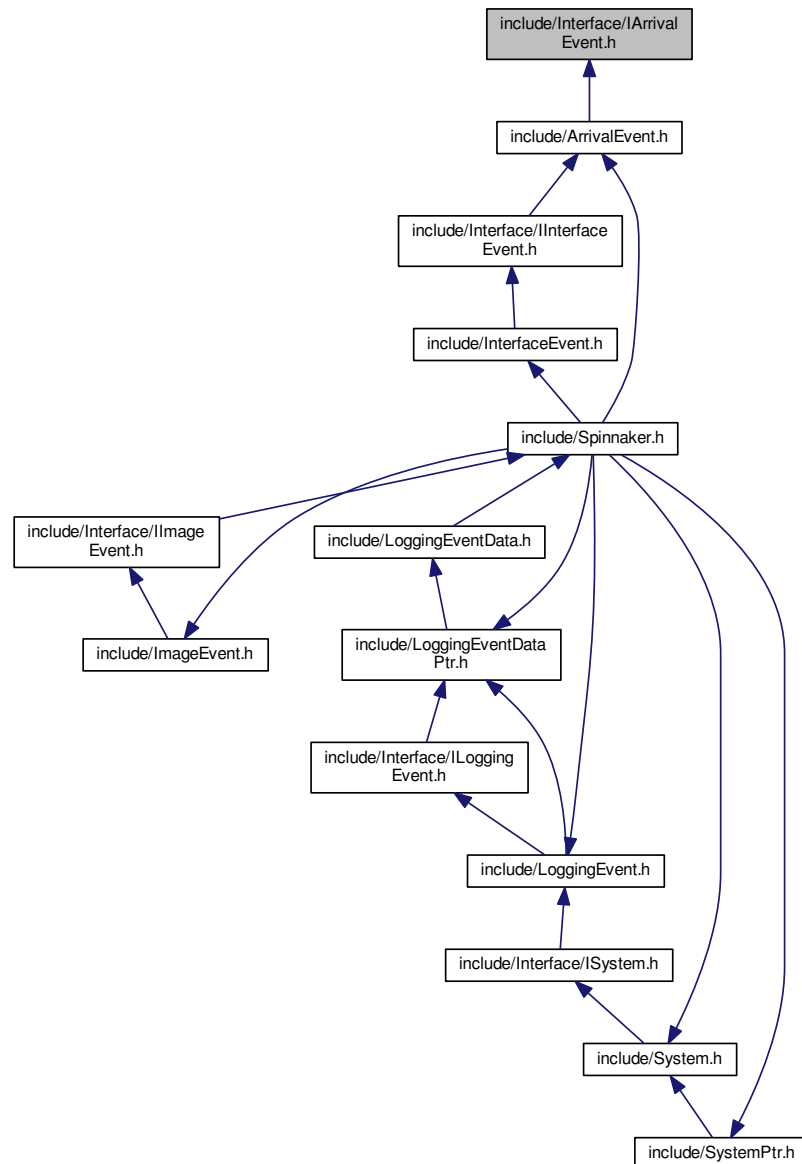
## 11.20 include/Interface/IArrivalEvent.h File Reference

Include dependency graph for IArrivalEvent.h:





This graph shows which files directly or indirectly include this file:



## Classes

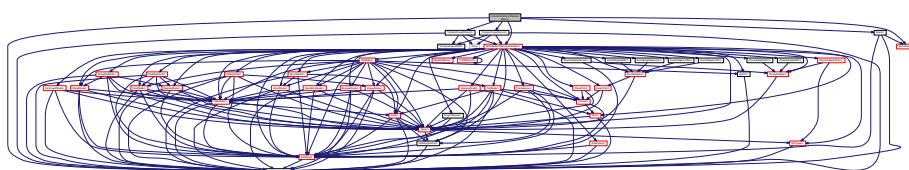
- class [IArrivalEvent](#)

## Namespaces

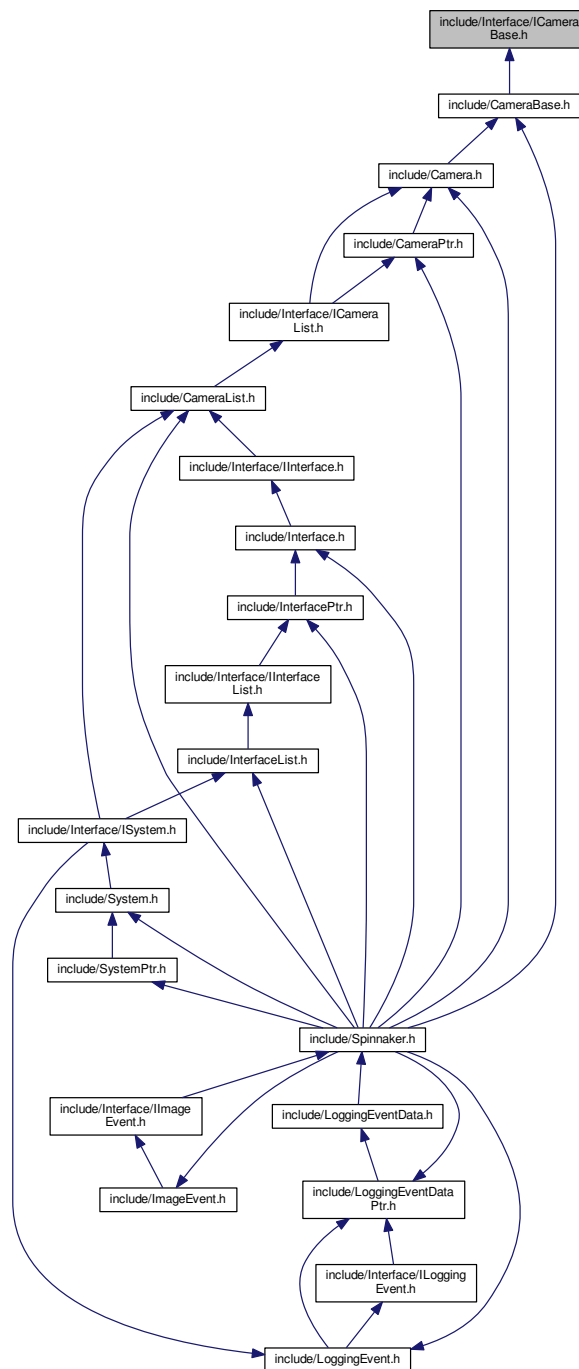
- [Spinnaker](#)

## 11.21 include/Interface/ICameraBase.h File Reference

Include dependency graph for ICameraBase.h:



This graph shows which files directly or indirectly include this file:



## Classes

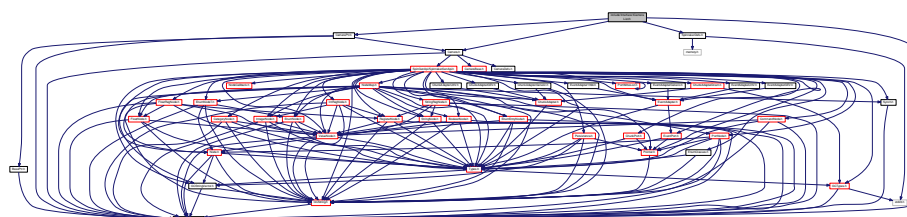
- class [ICameraBase](#)  
The interface file for base class for the camera object.

## Namespaces

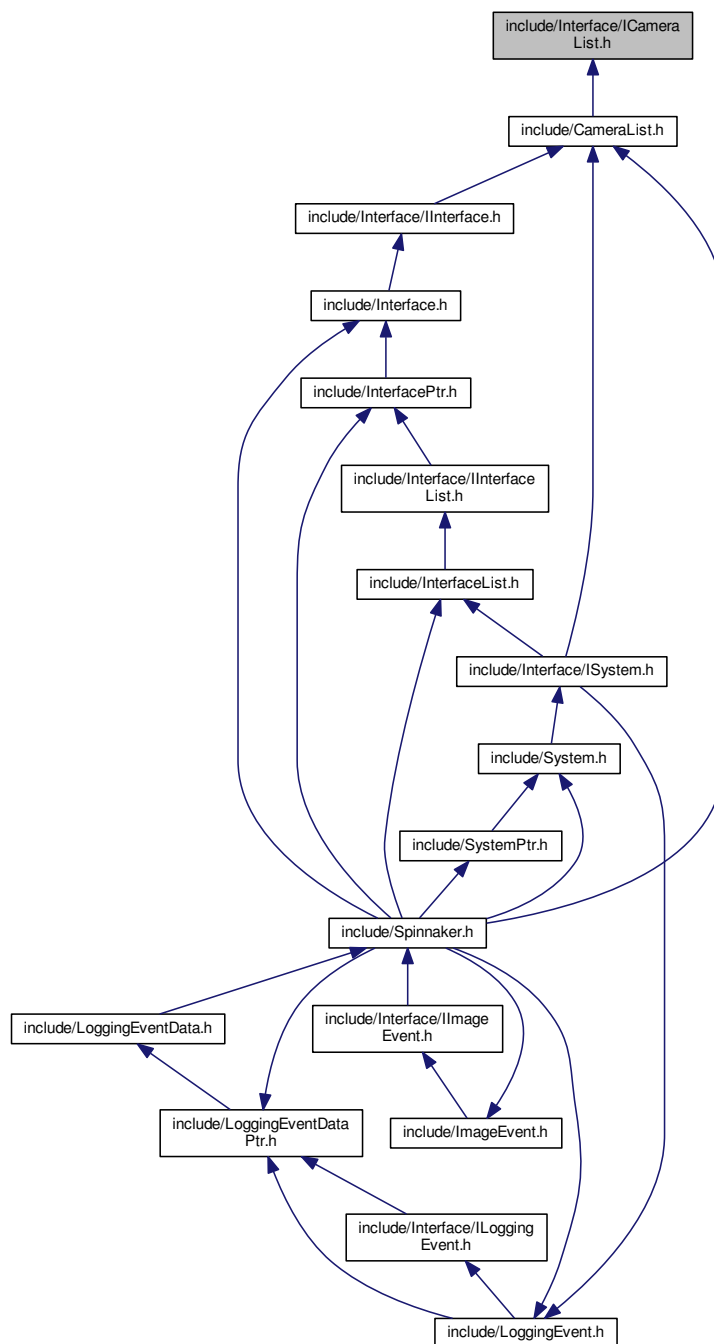
- [Spinnaker](#)

## 11.22 include/Interface/ICameraList.h File Reference

Include dependency graph for ICameraList.h:



This graph shows which files directly or indirectly include this file:



## Classes

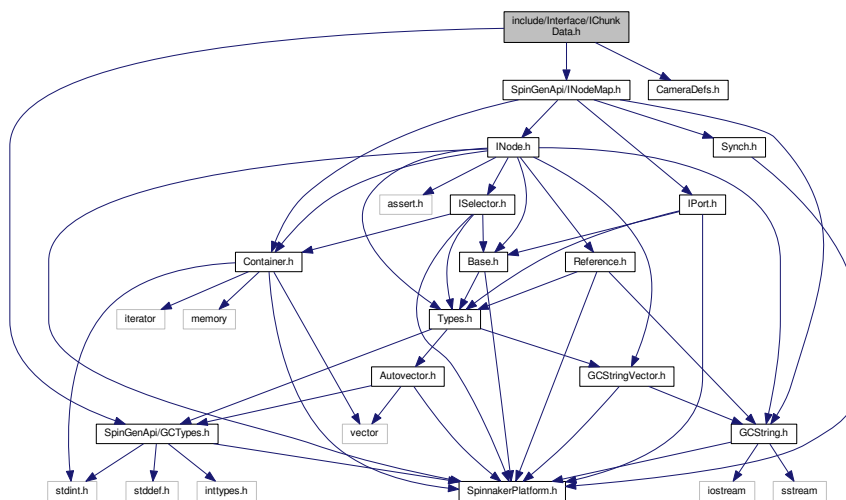
- class [ICameraList](#)  
Used to hold a list of camera objects.

## Namespaces

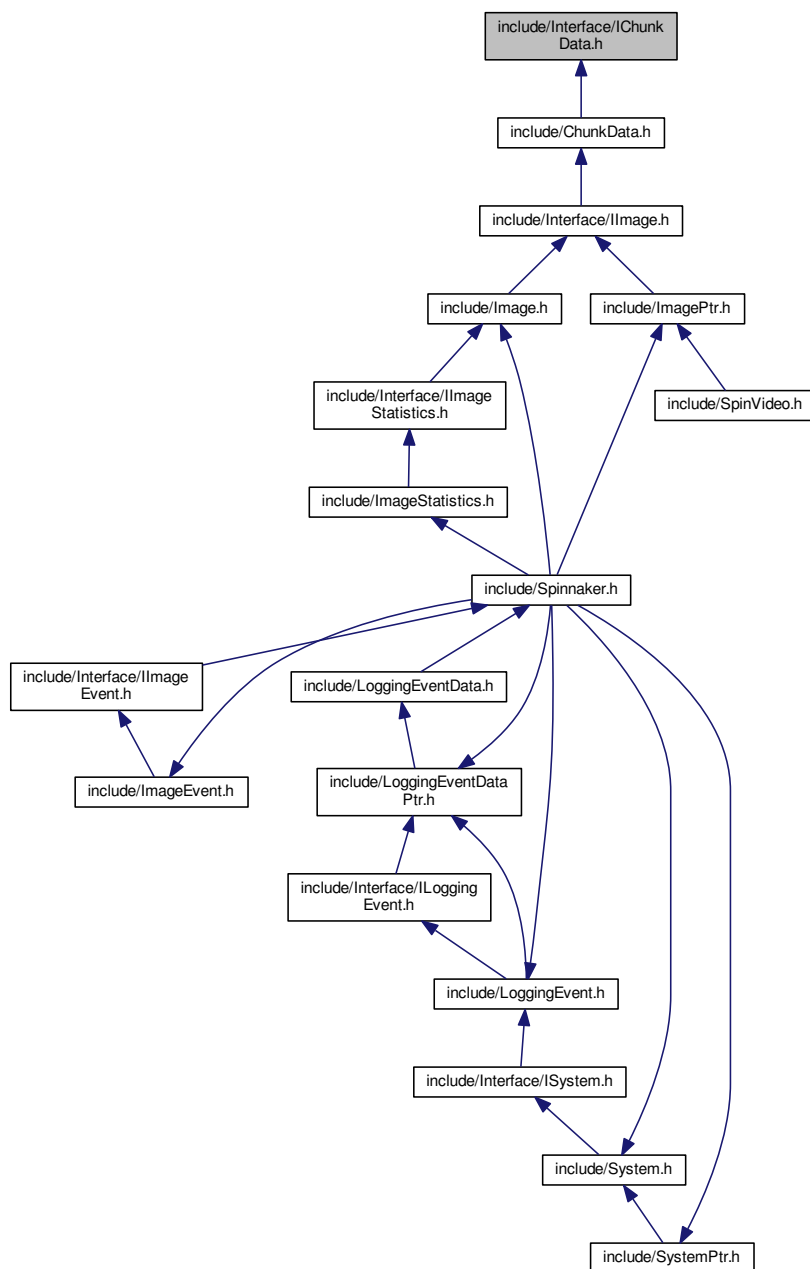
- [Spinnaker](#)

## 11.23 include/Interface/IChunkData.h File Reference

Include dependency graph for IChunkData.h:



This graph shows which files directly or indirectly include this file:



## Classes

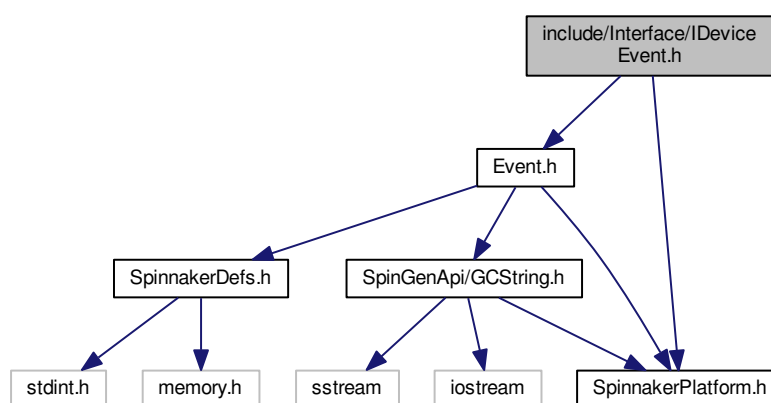
- class [IChunkData](#)  
The *Interface* file for *ChunkData*.

## Namespaces

- [Spinnaker](#)

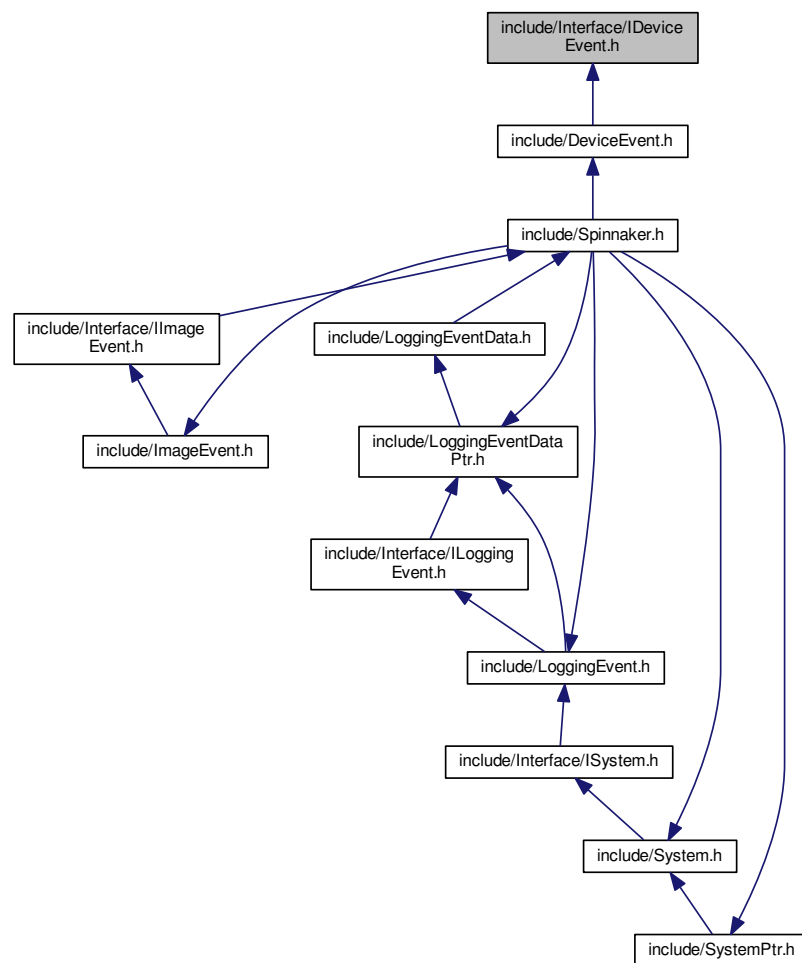
## 11.24 include/Interface/IDeviceEvent.h File Reference

Include dependency graph for IDeviceEvent.h:





This graph shows which files directly or indirectly include this file:



## Classes

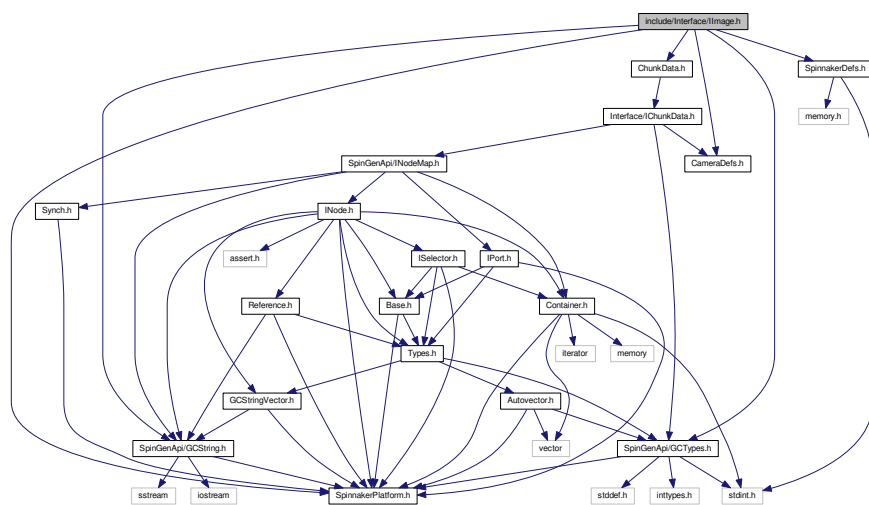
- class [IDeviceEvent](#)

## Namespaces

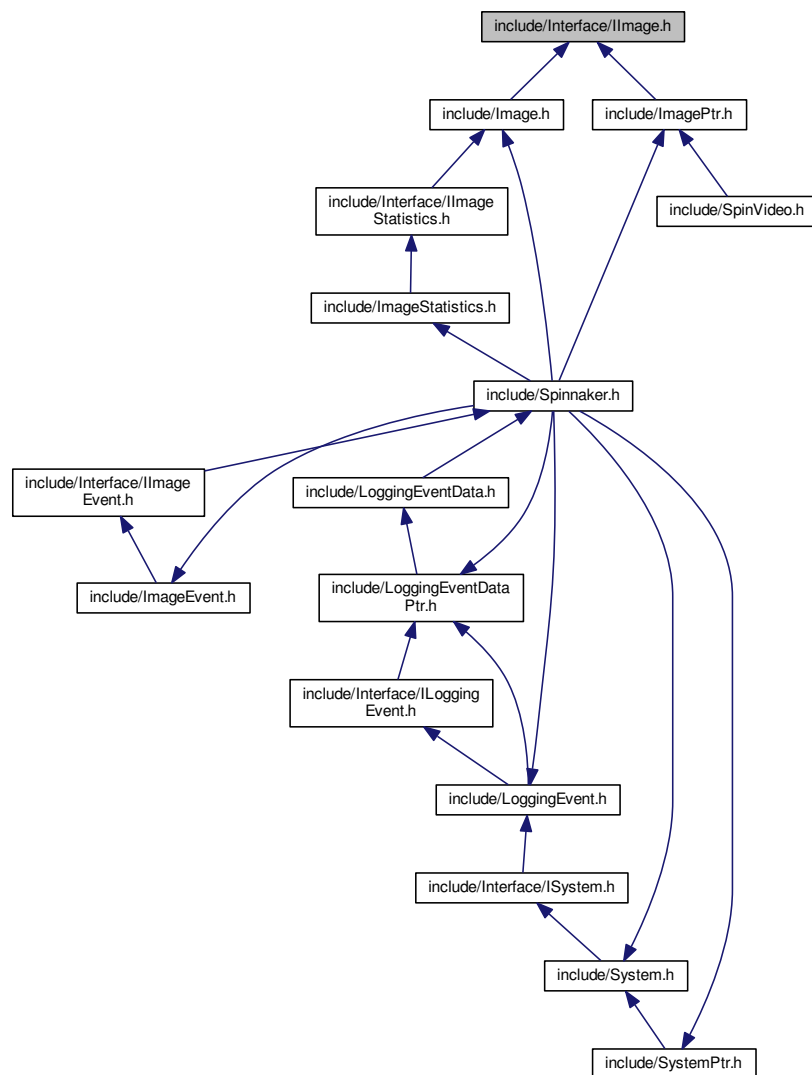
- [Spinnaker](#)

## 11.25 include/Interface/IImage.h File Reference

Include dependency graph for IImage.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [IImage](#)

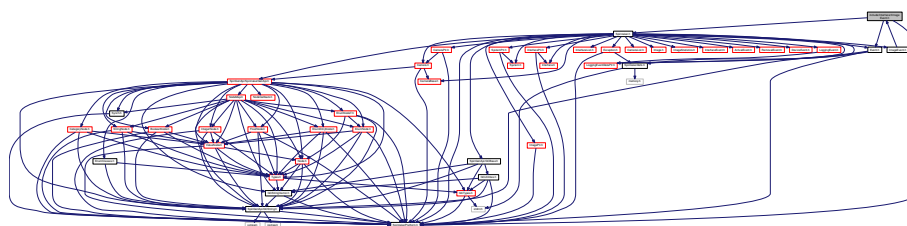
The interface file for [Image](#).

## Namespaces

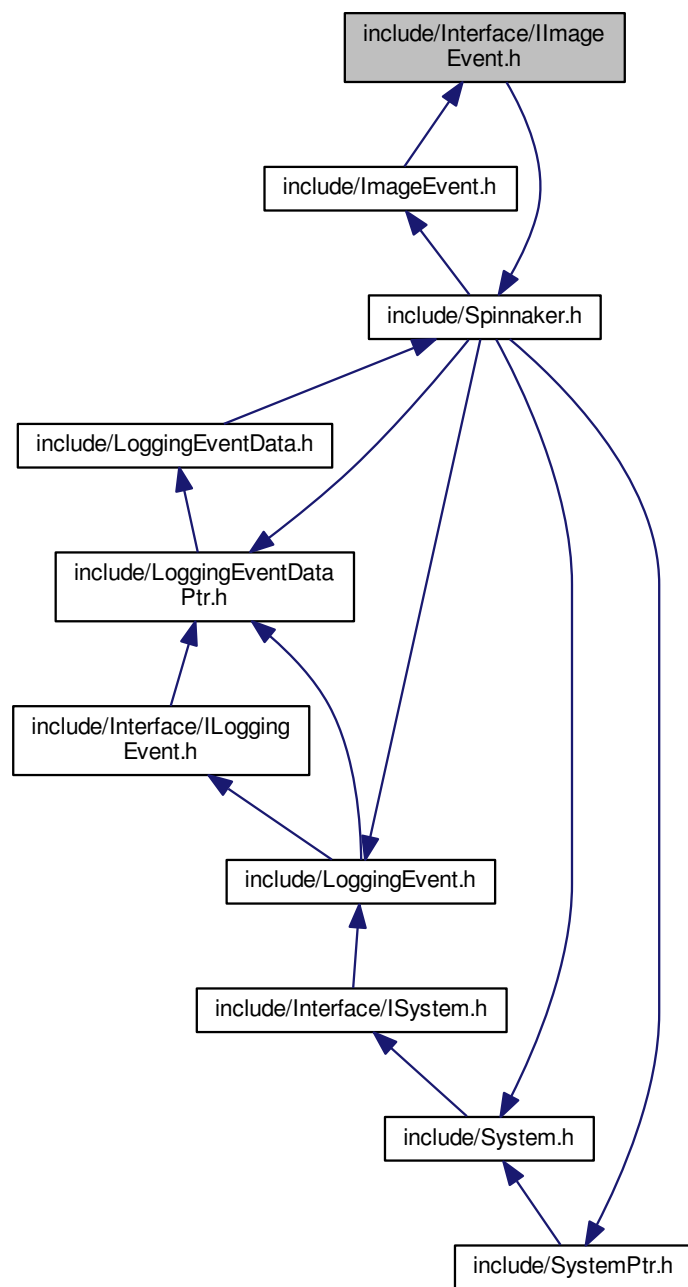
- [Spinnaker](#)

## 11.26 include/Interface/ImageEvent.h File Reference

Include dependency graph for ImageEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

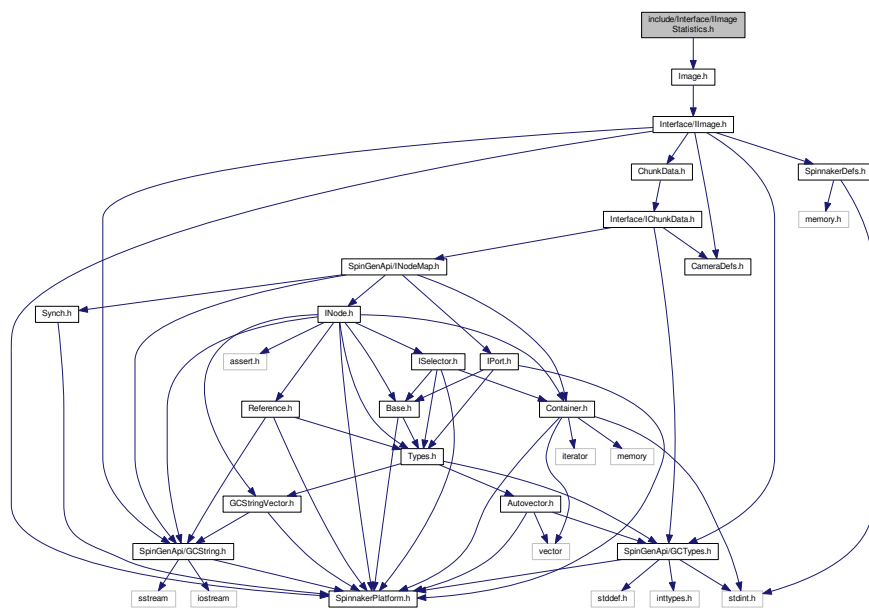
- class [IImageEvent](#)

## Namespaces

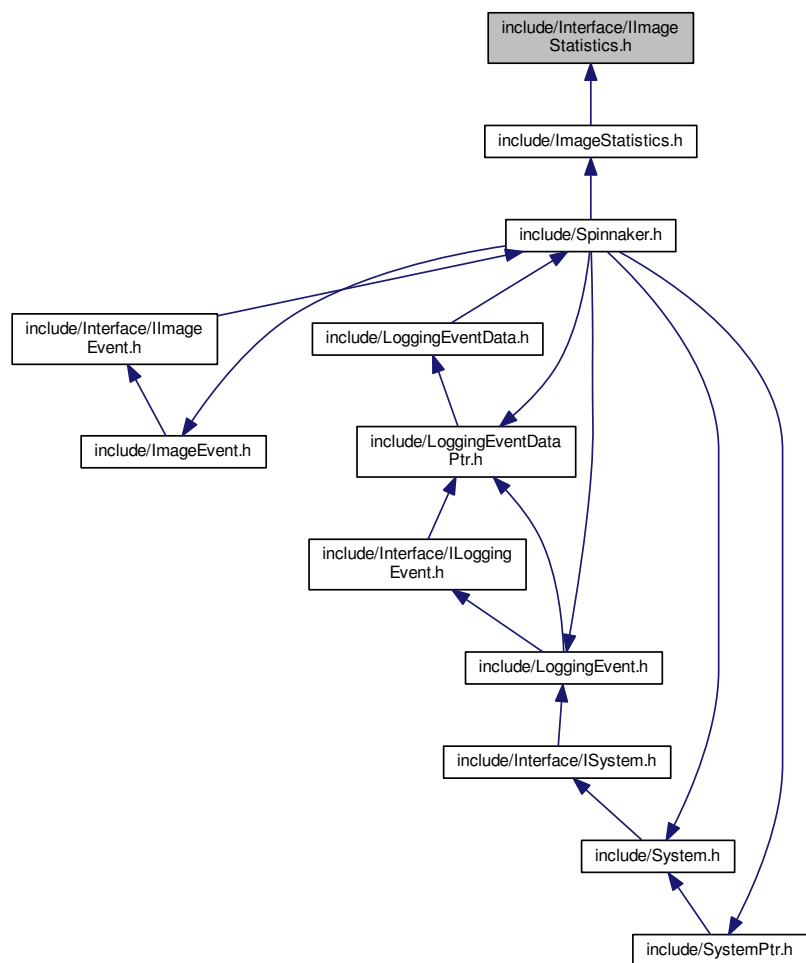
- [Spinnaker](#)

## 11.27 include/Interface/IImageStatistics.h File Reference

Include dependency graph for IImageStatistics.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [IImageStatistics](#)

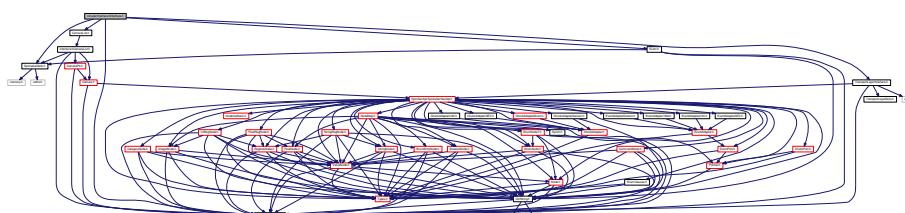
*The interface file for image statistics.*

## Namespaces

- [Spinnaker](#)

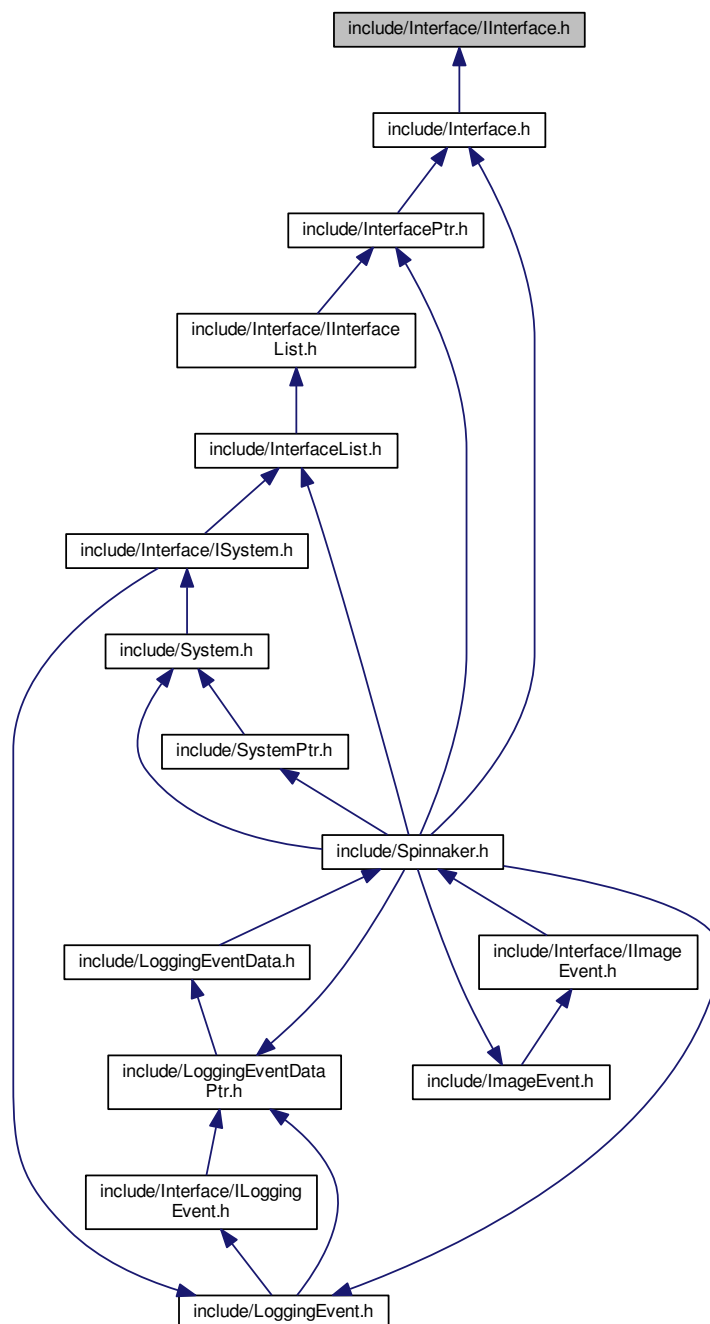
## 11.28 include/Interface/Interface.h File Reference

Include dependency graph for IInterface.h:





This graph shows which files directly or indirectly include this file:



## Classes

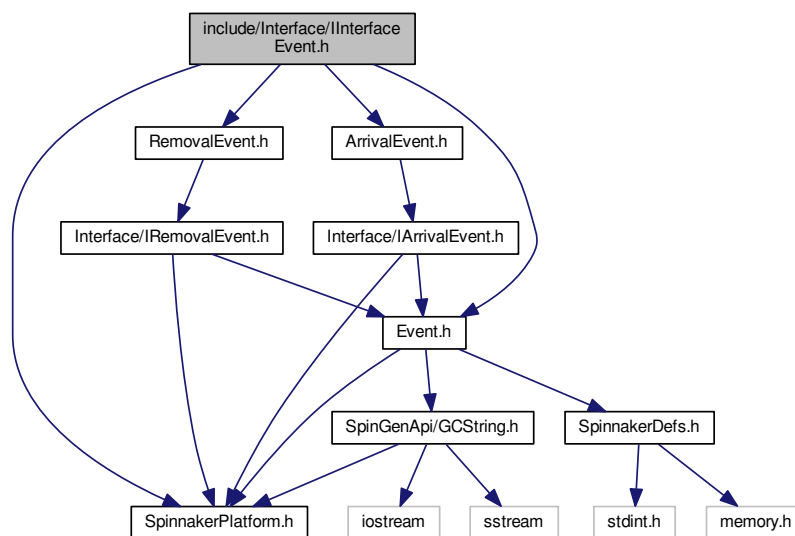
- class [IInterface](#)  
The interface file for [IInterface](#).

## Namespaces

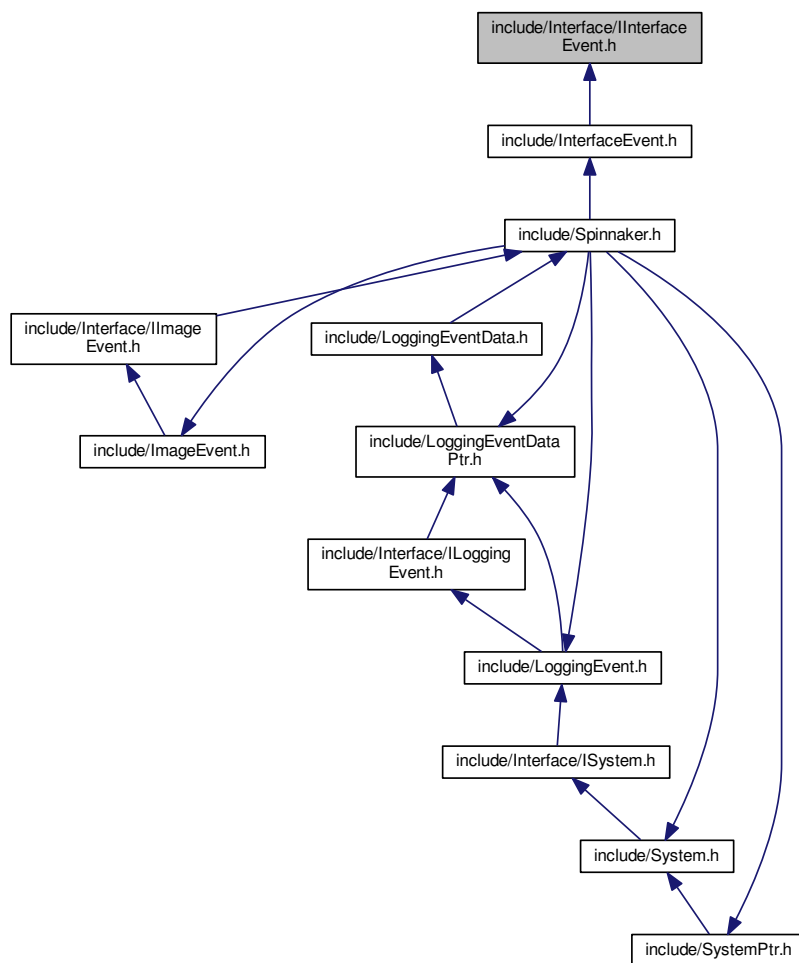
- [Spinnaker](#)

## 11.29 include/Interface/InterfaceEvent.h File Reference

Include dependency graph for InterfaceEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

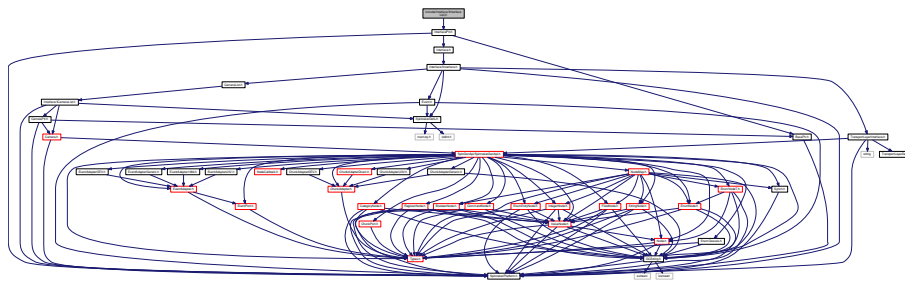
- class `InterfaceEvent`

## Namespaces

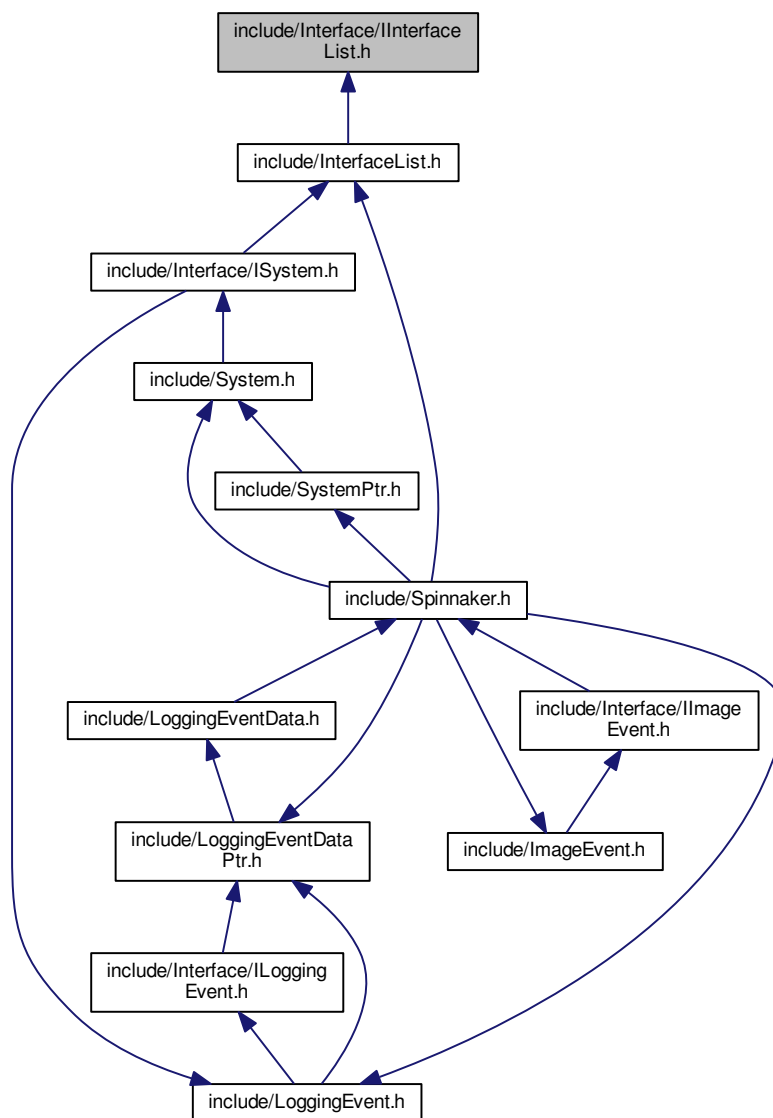
- Spinnaker

### 11.30 include/Interface/IInterfaceList.h File Reference

Include dependency graph for IInterfaceList.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [IInterfaceList](#)

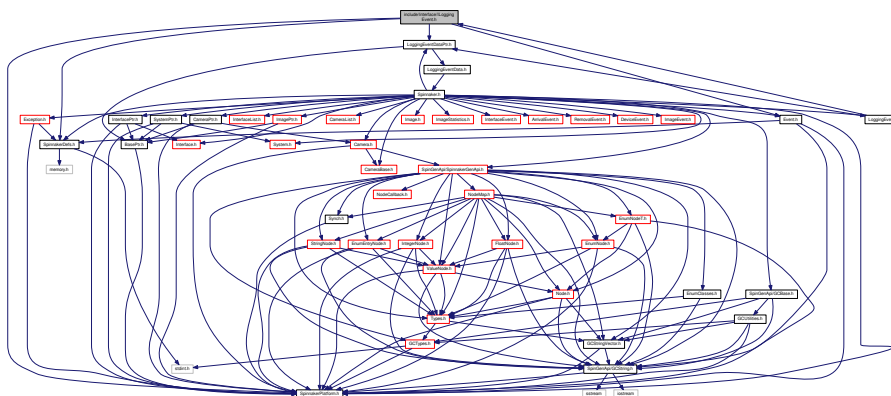
The interface file for [IInterfaceList](#) class.

## Namespaces

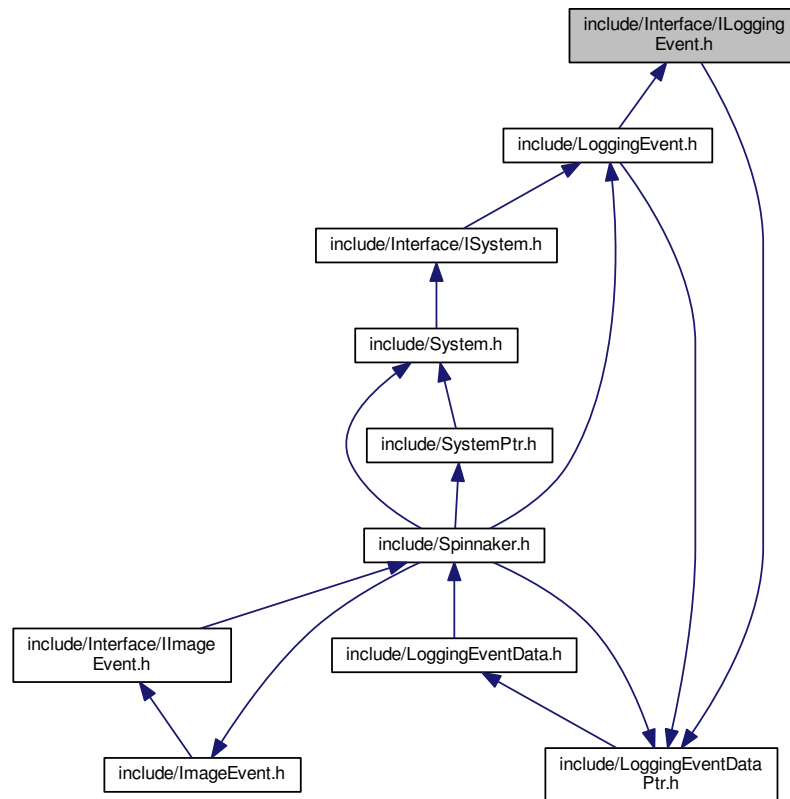
- [Spinnaker](#)

## 11.31 include/Interface/ILoggingEvent.h File Reference

Include dependency graph for ILoggingEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

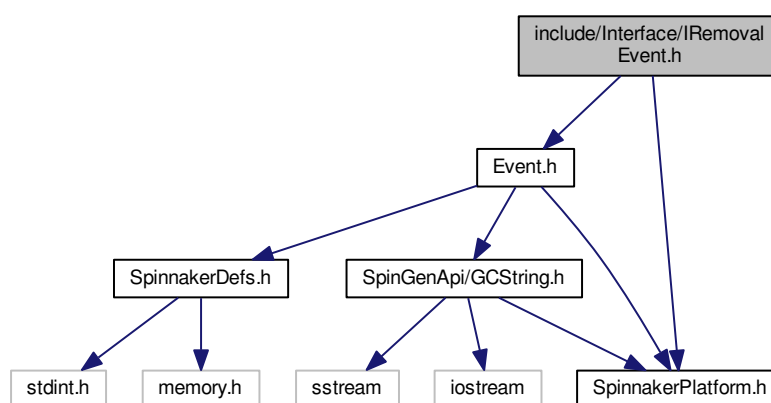
- class [ILoggingEvent](#)

## Namespaces

- [Spinnaker](#)

## 11.32 include/Interface/IRemovalEvent.h File Reference

Include dependency graph for IRemovalEvent.h:

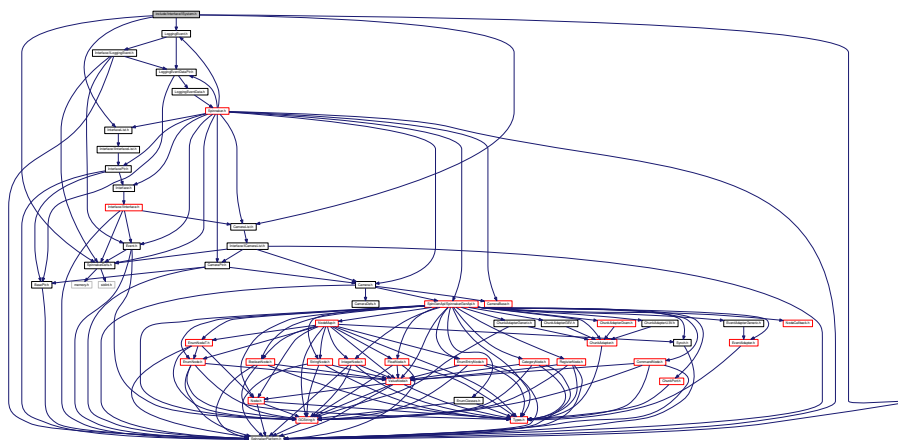




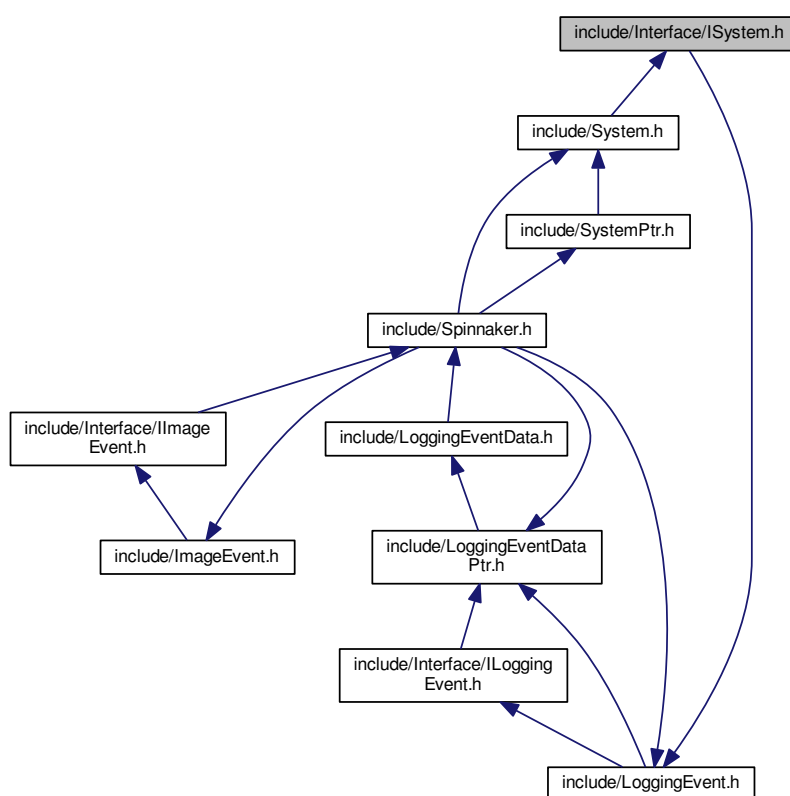


## 11.33 include/Interface/ISystem.h File Reference

Include dependency graph for ISystem.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [ISystem](#)

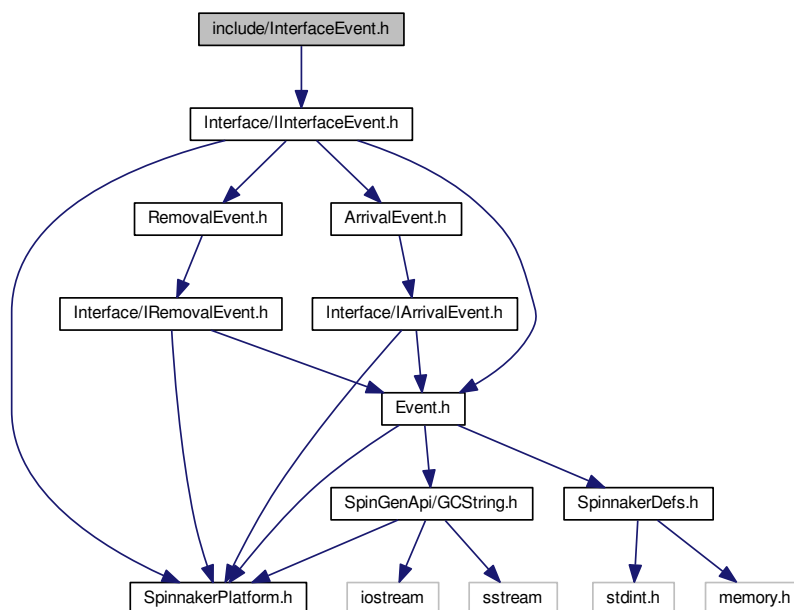
The interface file for [System](#).

## Namespaces

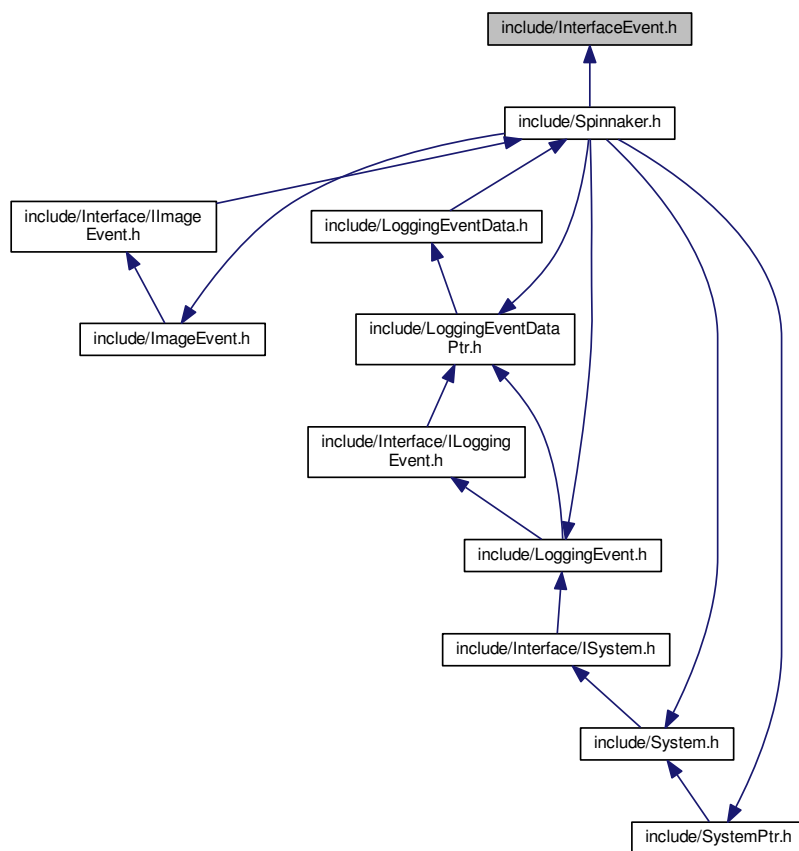
- [Spinnaker](#)

### 11.34 include/InterfaceEvent.h File Reference

Include dependency graph for InterfaceEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [InterfaceEvent](#)

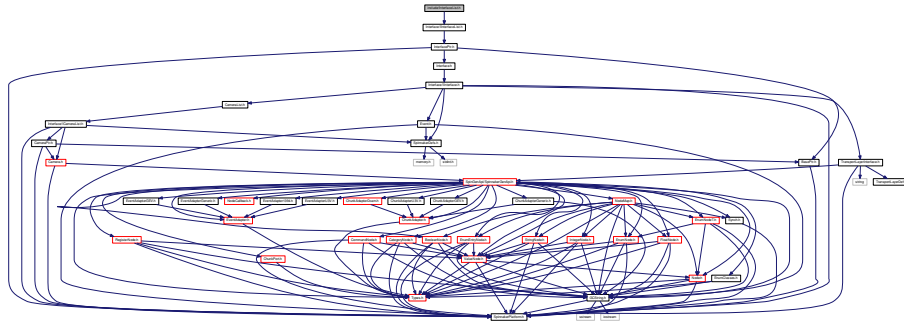
*A handler to device arrival and removal events on all interfaces.*

## Namespaces

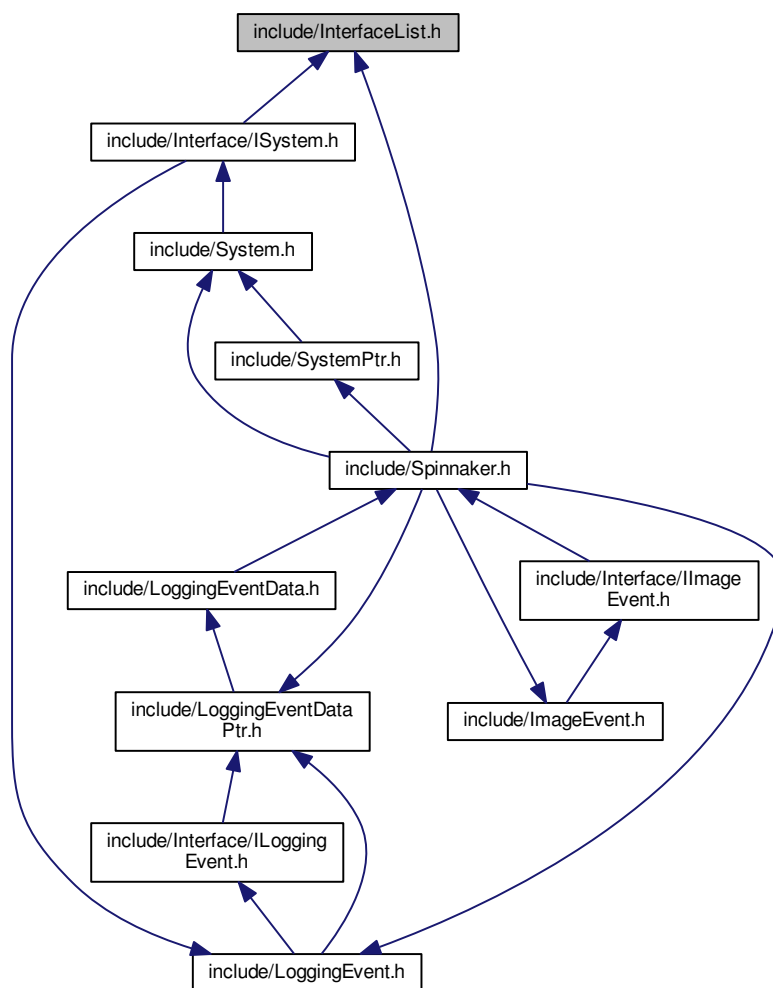
- [Spinnaker](#)

## 11.35 include/InterfaceList.h File Reference

Include dependency graph for InterfaceList.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [InterfaceList](#)

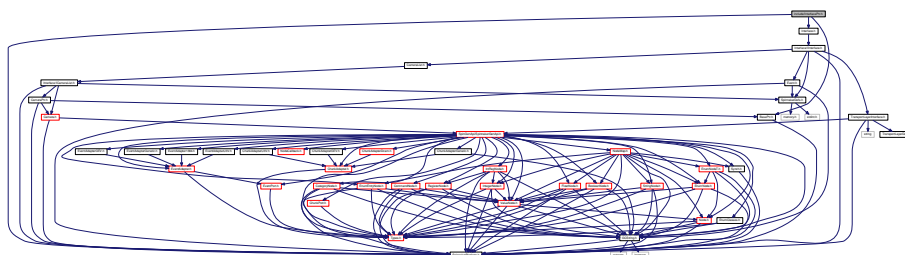
*A list of the available interfaces on the system.*

## Namespaces

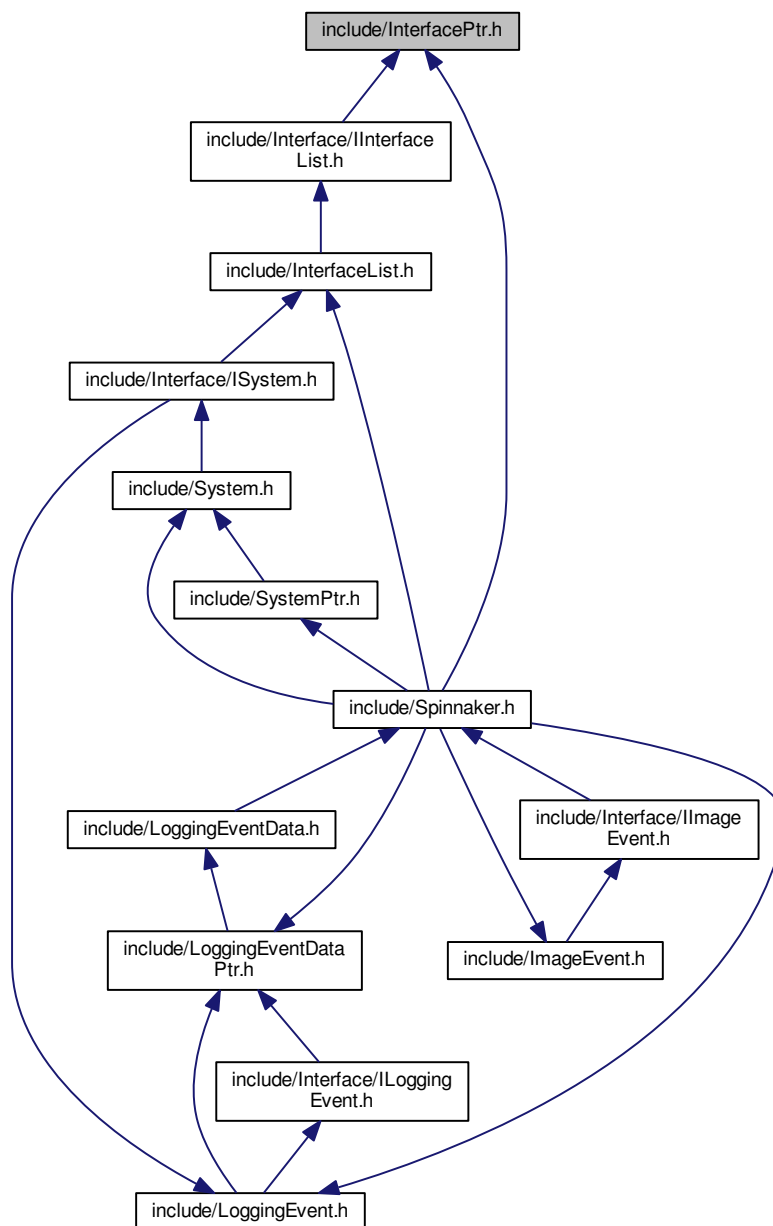
- [Spinnaker](#)

## 11.36 include/InterfacePtr.h File Reference

Include dependency graph for InterfacePtr.h:



This graph shows which files directly or indirectly include this file:



## Classes

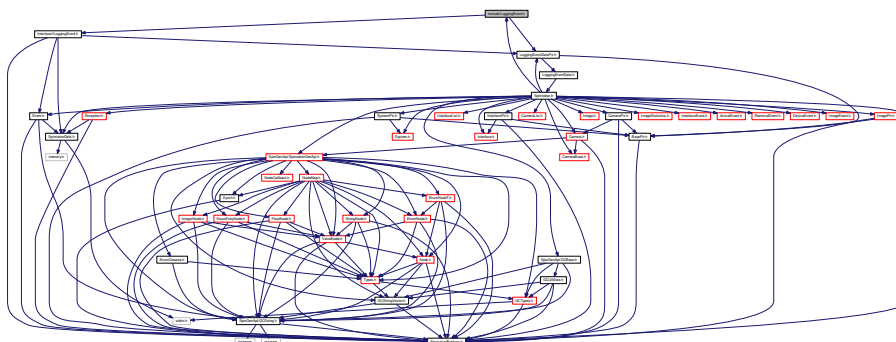
- class [InterfacePtr](#)  
A reference tracked pointer to the interface object.

## Namespaces

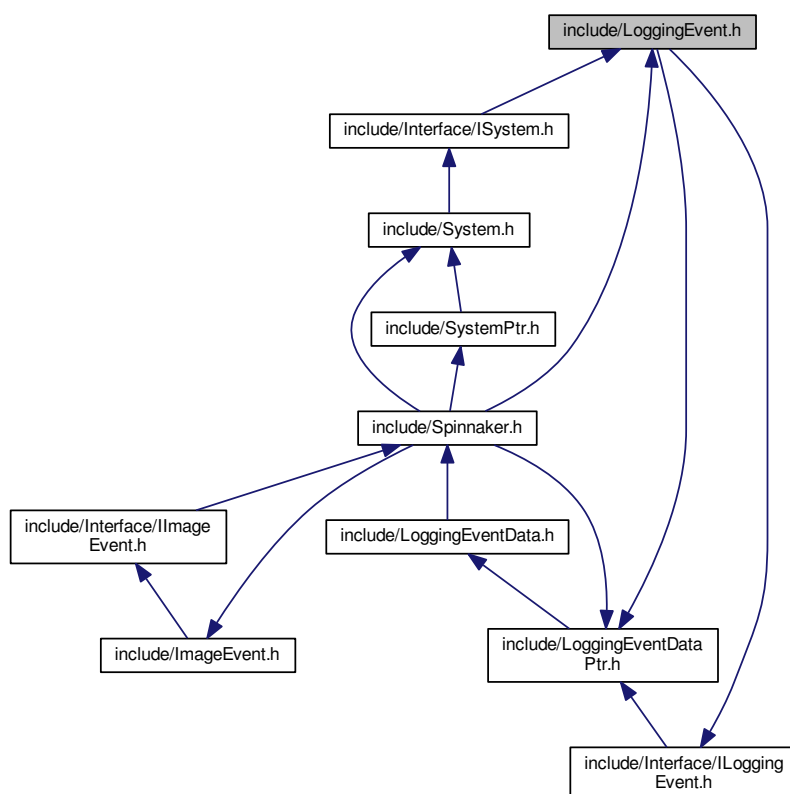
- [Spinnaker](#)

## 11.37 include/LoggingEvent.h File Reference

Include dependency graph for LoggingEvent.h:



This graph shows which files directly or indirectly include this file:



### Classes

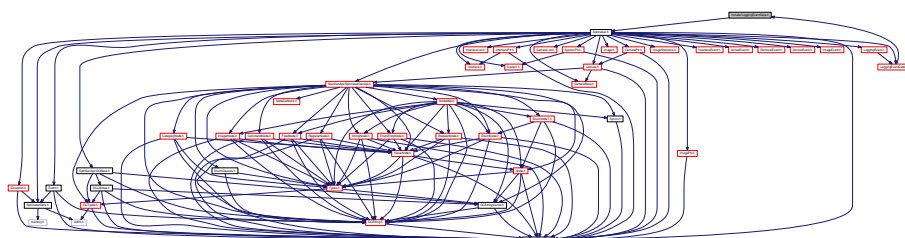
- class [LoggingEvent](#)  
An event handler for capturing the device logging event.

## Namespaces

- [Spinnaker](#)

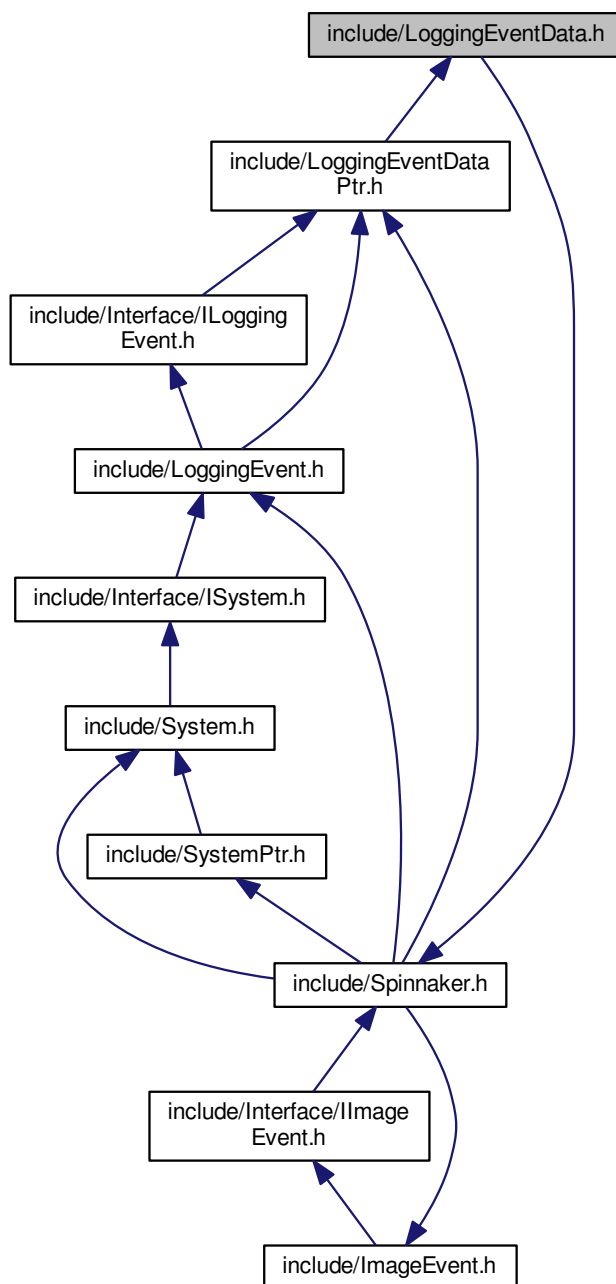
### 11.38 include/LoggingEventData.h File Reference

Include dependency graph for LoggingEventData.h:





This graph shows which files directly or indirectly include this file:



## Classes

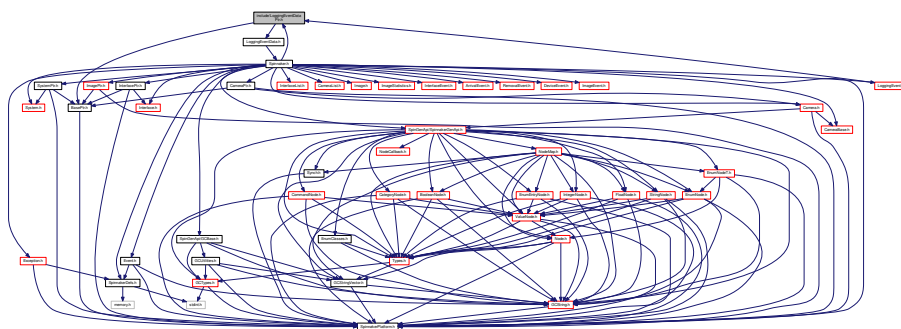
- class [LoggingEventData](#)  
The *LoggingEventData* object.

## Namespaces

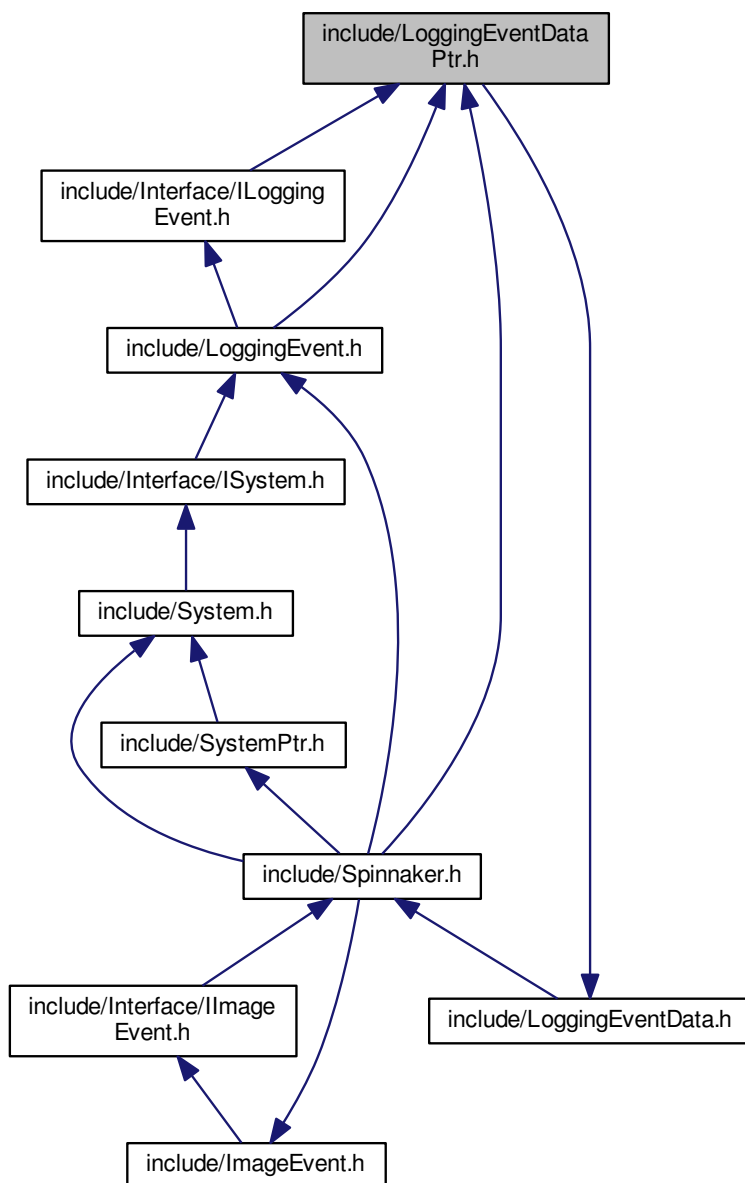
- [Spinnaker](#)

### 11.39 include/LoggingEventDataPtr.h File Reference

Include dependency graph for LoggingEventDataPtr.h:



This graph shows which files directly or indirectly include this file:



## Classes

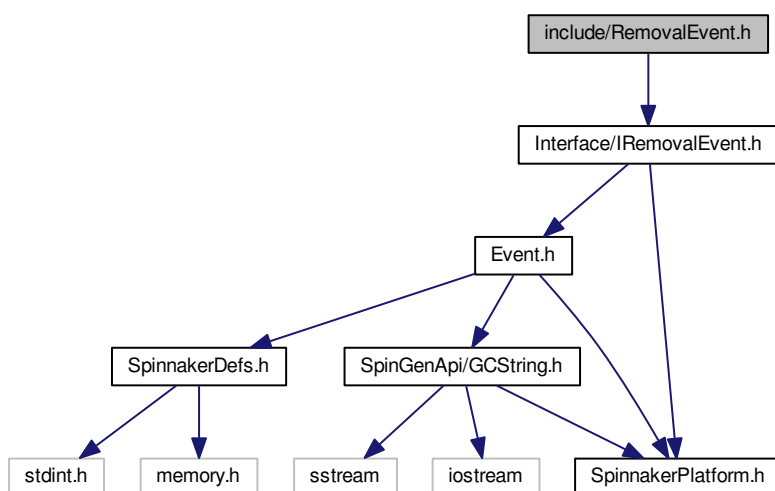
- class [LoggingEventDataPtr](#)  
A reference tracked pointer to the [LoggingEvent](#) object.

## Namespaces

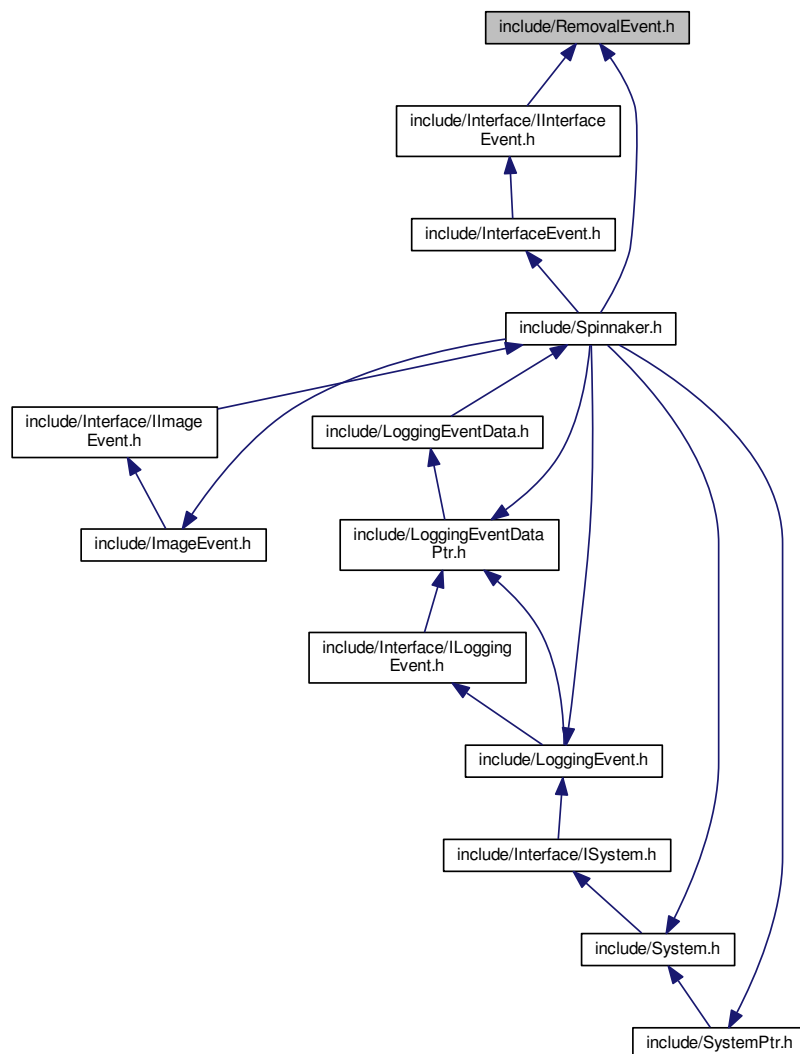
- [Spinnaker](#)

## 11.40 include/RemovalEvent.h File Reference

Include dependency graph for RemovalEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [RemovalEvent](#)

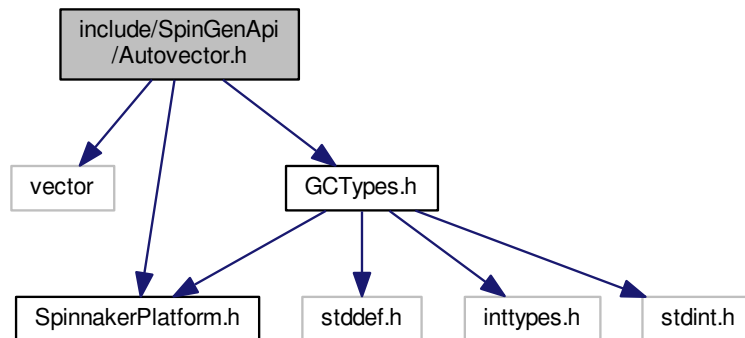
*An event handler for capturing the device removal event.*

## Namespaces

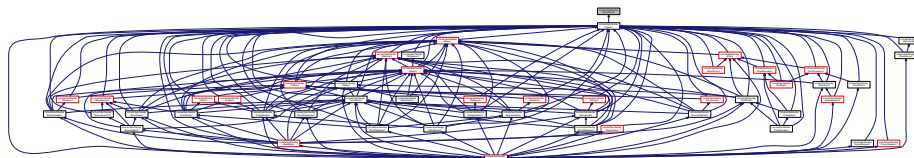
- [Spinnaker](#)

## 11.41 include/SpinGenApi/Autovector.h File Reference

Include dependency graph for Autovector.h:



This graph shows which files directly or indirectly include this file:



### Classes

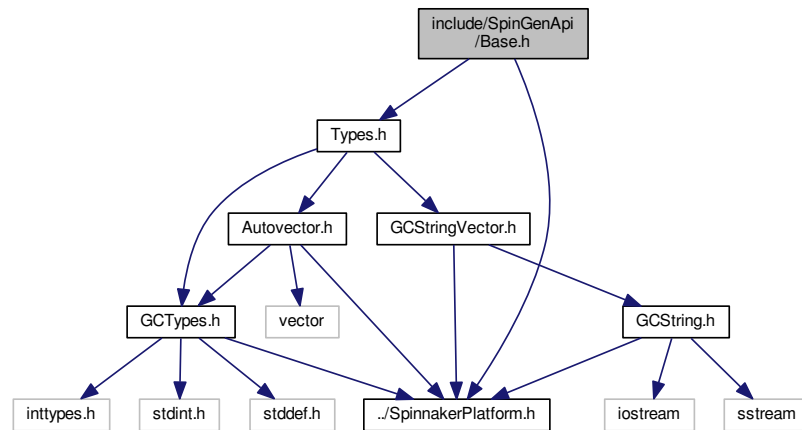
- class [int64\\_autovector\\_t](#)  
*Vector of integers with reference counting.*
- class [double\\_autovector\\_t](#)  
*Vector of doubles with reference counting.*

### Namespaces

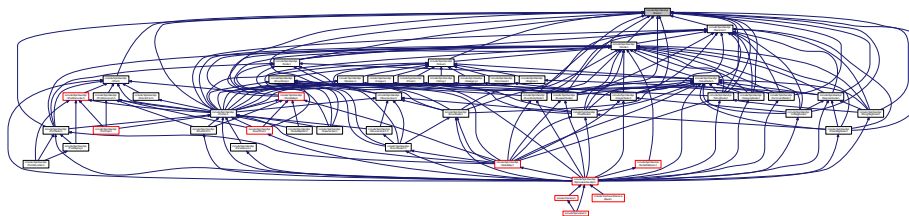
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.42 include/SpinGenApi/Base.h File Reference

Include dependency graph for Base.h:



This graph shows which files directly or indirectly include this file:



### Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

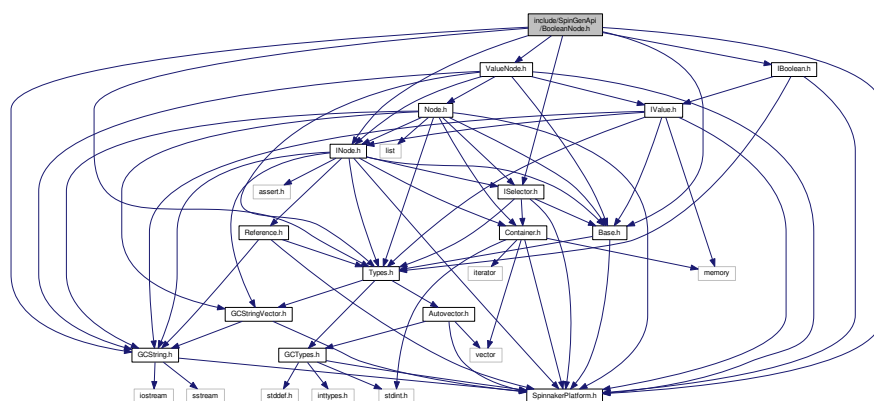
### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IBase](#)

*Base interface common to all nodes.*

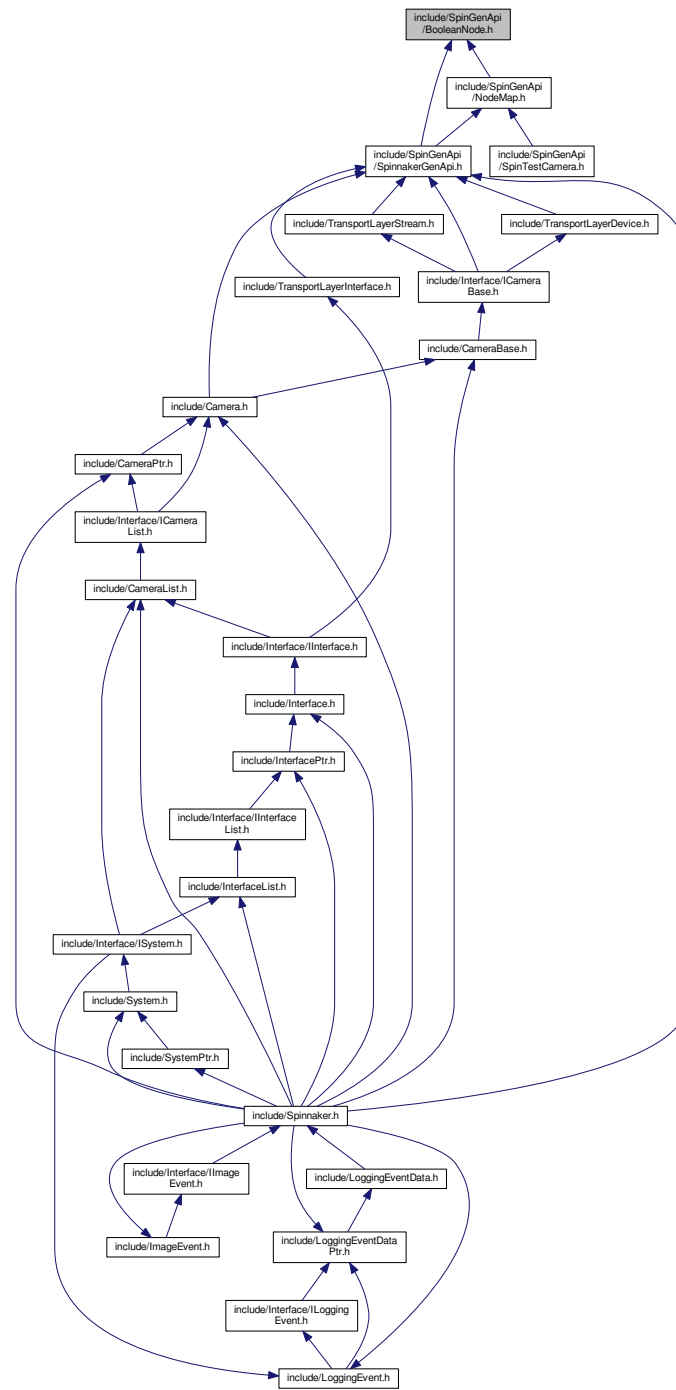
## 11.43 include/SpinGenApi/BooleanNode.h File Reference

Include dependency graph for BooleanNode.h:





This graph shows which files directly or indirectly include this file:

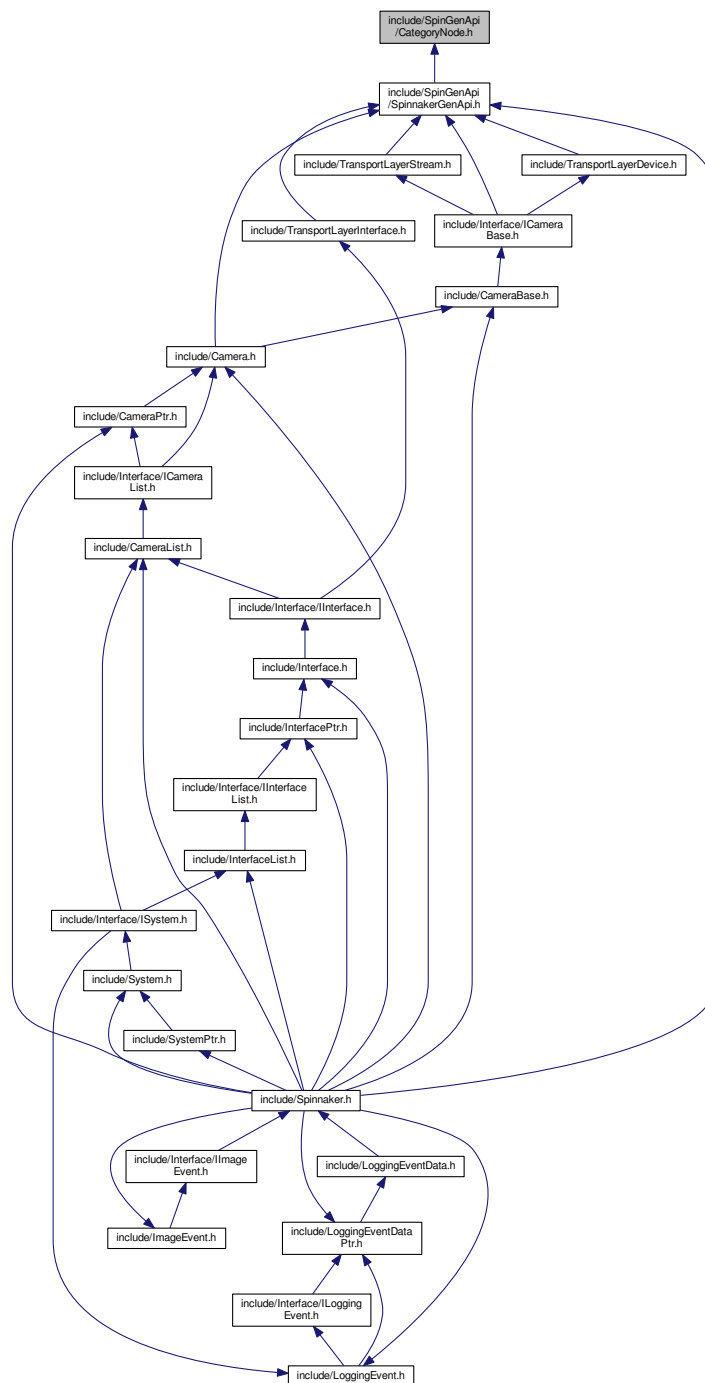


## Classes

- class [BooleanNode](#)  
*Interface for string properties.*



This graph shows which files directly or indirectly include this file:



## Classes

- class [CategoryNode](#)  
*Interface for string properties.*

## Namespaces

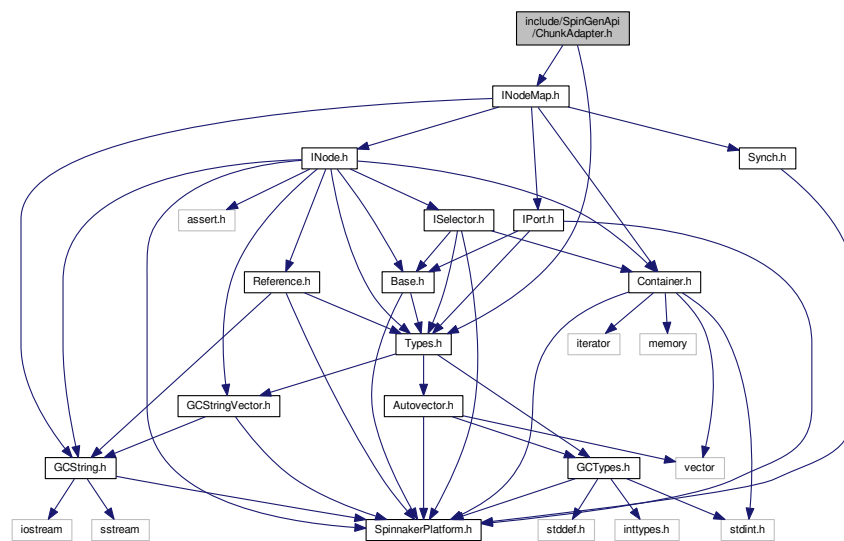
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

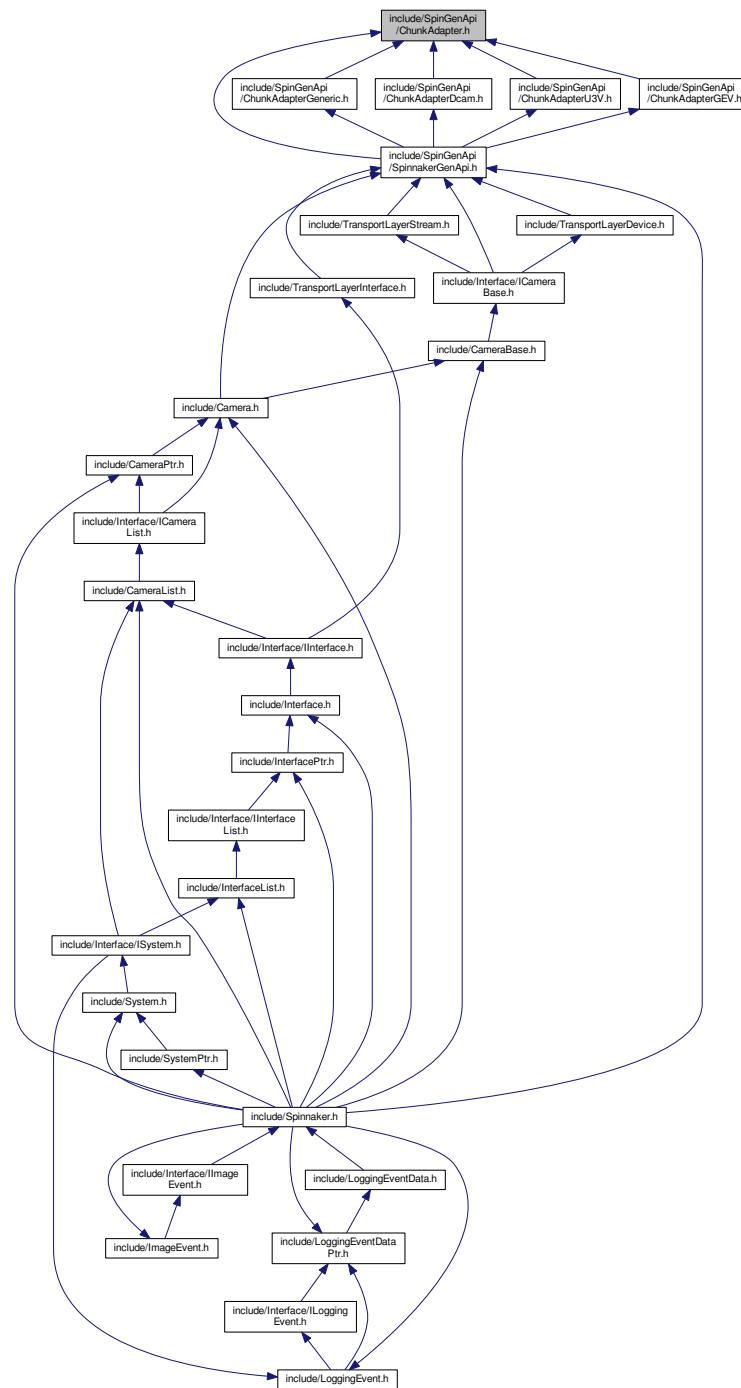
- typedef CategoryNode [CCategoryRef](#)

## 11.45 include/SpinGenApi/ChunkAdapter.h File Reference

Include dependency graph for ChunkAdapter.h:



This graph shows which files directly or indirectly include this file:



## Classes

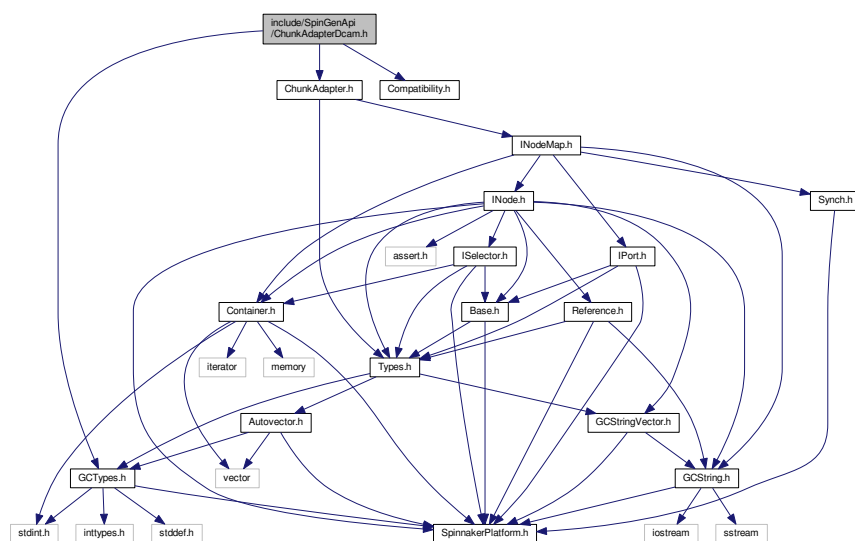
- struct [AttachStatistics\\_t](#)  
*Delivers information about the attached chunks and nodes.*
- class [CChunkAdapter](#)  
*Connects a chunked buffer to a node map.*

## Namespaces

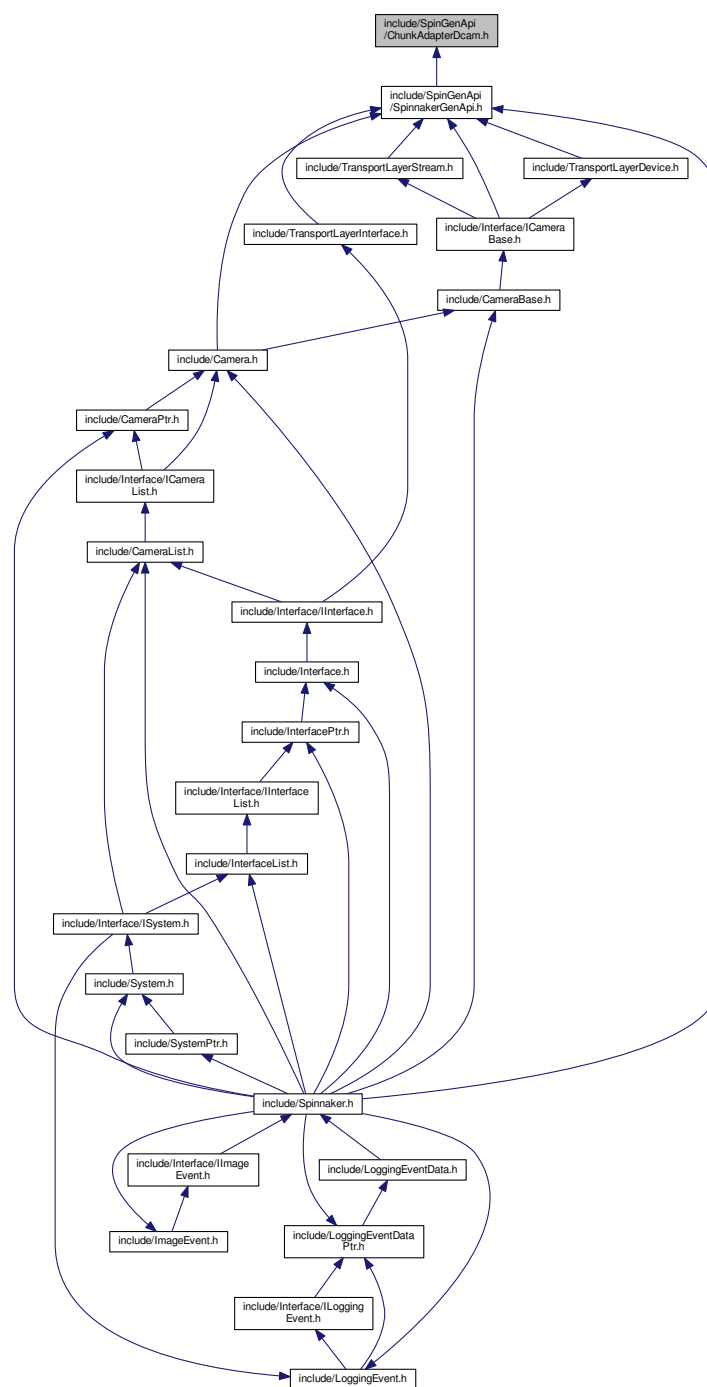
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

### 11.46 include/SpinGenApi/ChunkAdapterDcam.h File Reference

Include dependency graph for ChunkAdapterDcam.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [DCAM\\_CHUNK\\_TRAILER](#)
- struct [DCAM\\_CHECKSUM](#)
- class [CChunkAdapterDcam](#)

*Connects a chunked DCAM buffer to a node map.*

## Namespaces

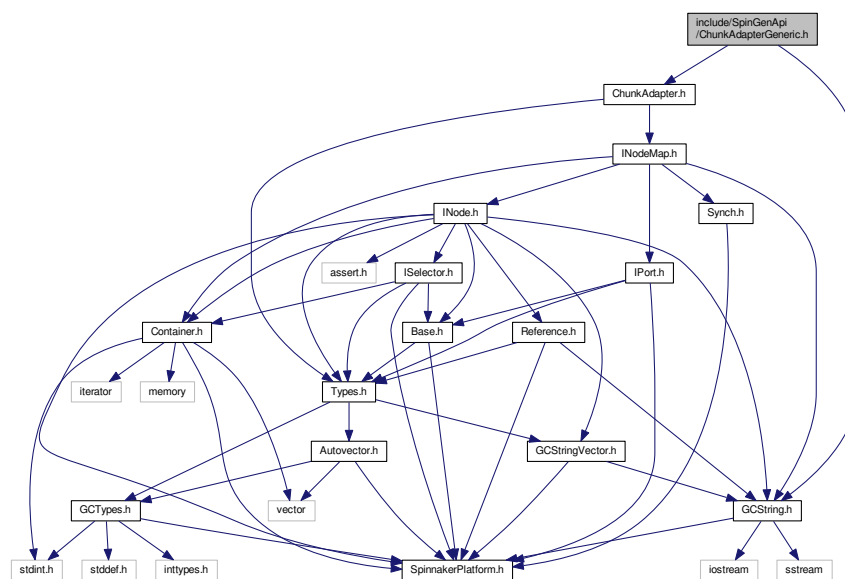
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

- void [SPINNAKER\\_API SET\\_GUID](#) (SPIN\_GUID &name, uint32\_t l, uint16\_t w1, uint16\_t w2, uint8\_t b1, uint8\_t b2, uint8\_t b3, uint8\_t b4, uint8\_t b5, uint8\_t b6, uint8\_t b7, uint8\_t b8)

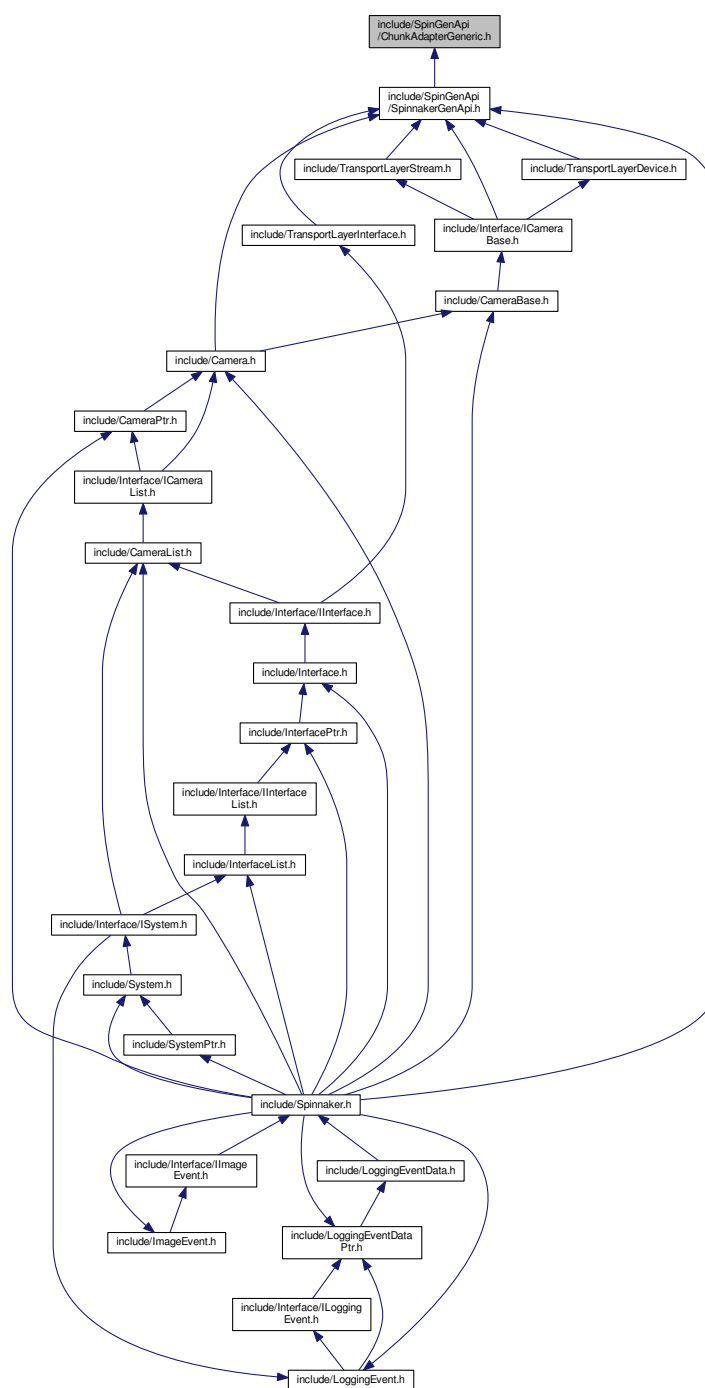
## 11.47 include/SpinGenApi/ChunkAdapterGeneric.h File Reference

Include dependency graph for ChunkAdapterGeneric.h:





This graph shows which files directly or indirectly include this file:



## Classes

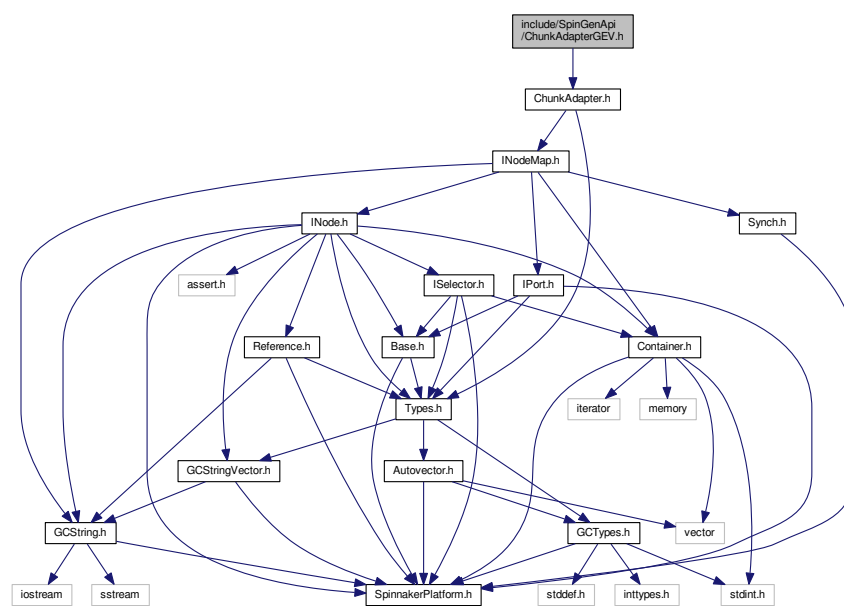
- struct [SingleChunkData\\_t](#)
- struct [SingleChunkDataStr\\_t](#)
- class [CChunkAdapterGeneric](#)

## Namespaces

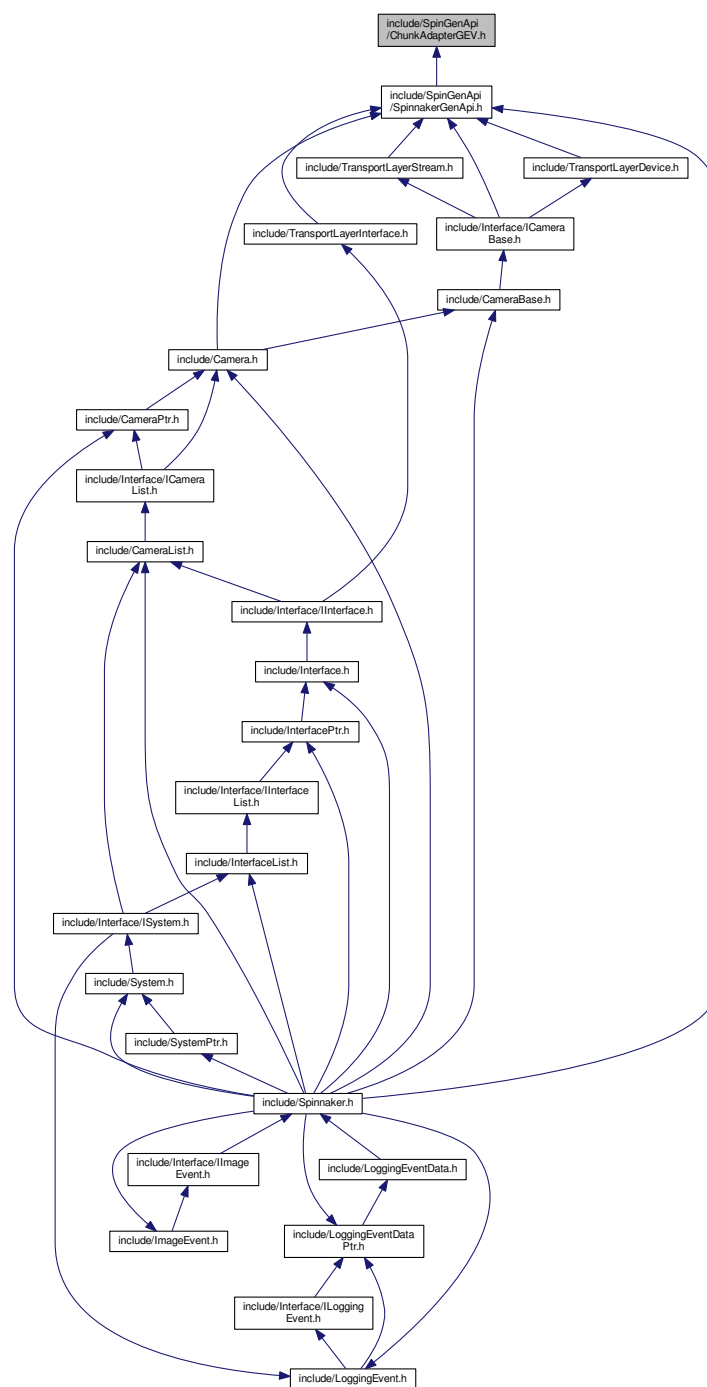
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.48 include/SpinGenApi/ChunkAdapterGEV.h File Reference

Include dependency graph for ChunkAdapterGEV.h:



This graph shows which files directly or indirectly include this file:



## Classes

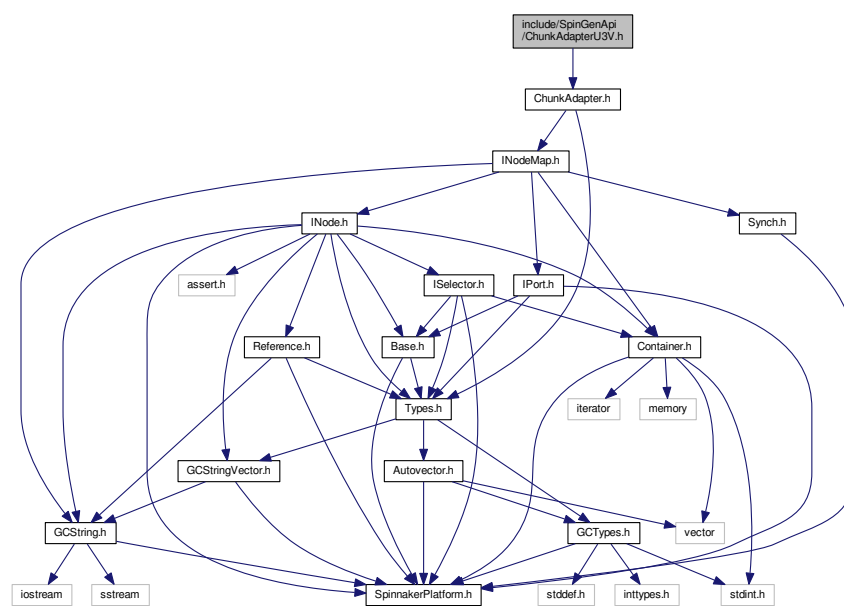
- struct [GVCP\\_CHUNK\\_TRAILER](#)  
header of a GVCP request packet
- class [CChunkAdapterGEV](#)  
Connects a chunked DCAM buffer to a node map.

## Namespaces

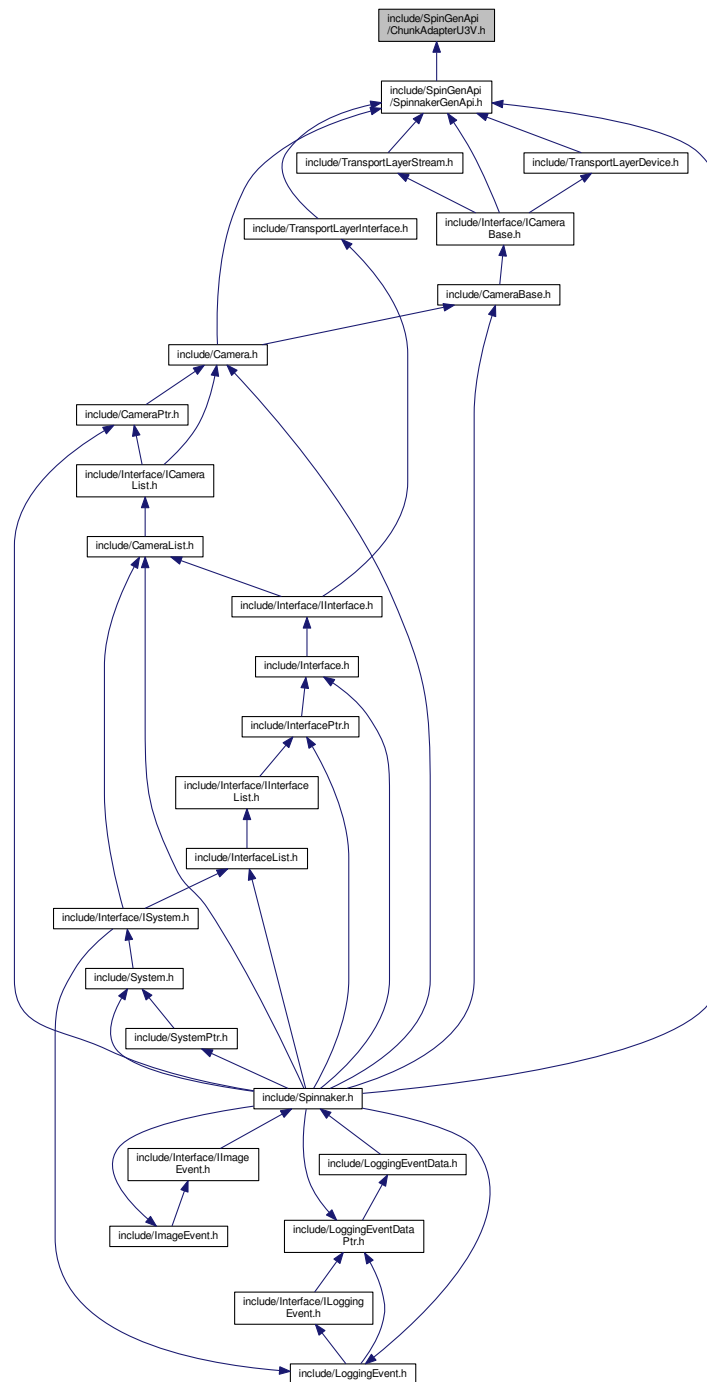
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.49 include/SpinGenApi/ChunkAdapterU3V.h File Reference

Include dependency graph for ChunkAdapterU3V.h:



This graph shows which files directly or indirectly include this file:



## Classes

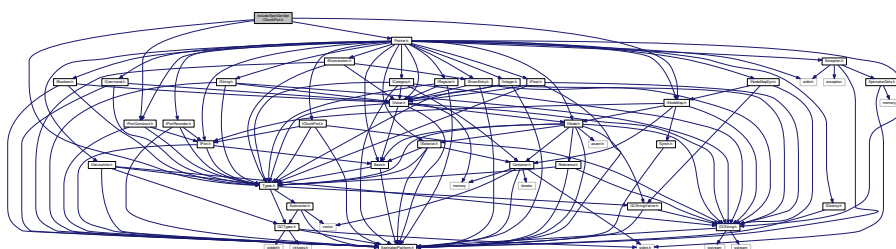
- struct [U3V\\_CHUNK\\_TRAILER](#)  
header of a GVCP request packet
- class [CChunkAdapterU3V](#)  
Connects a chunked U3V buffer to a node map.

## Namespaces

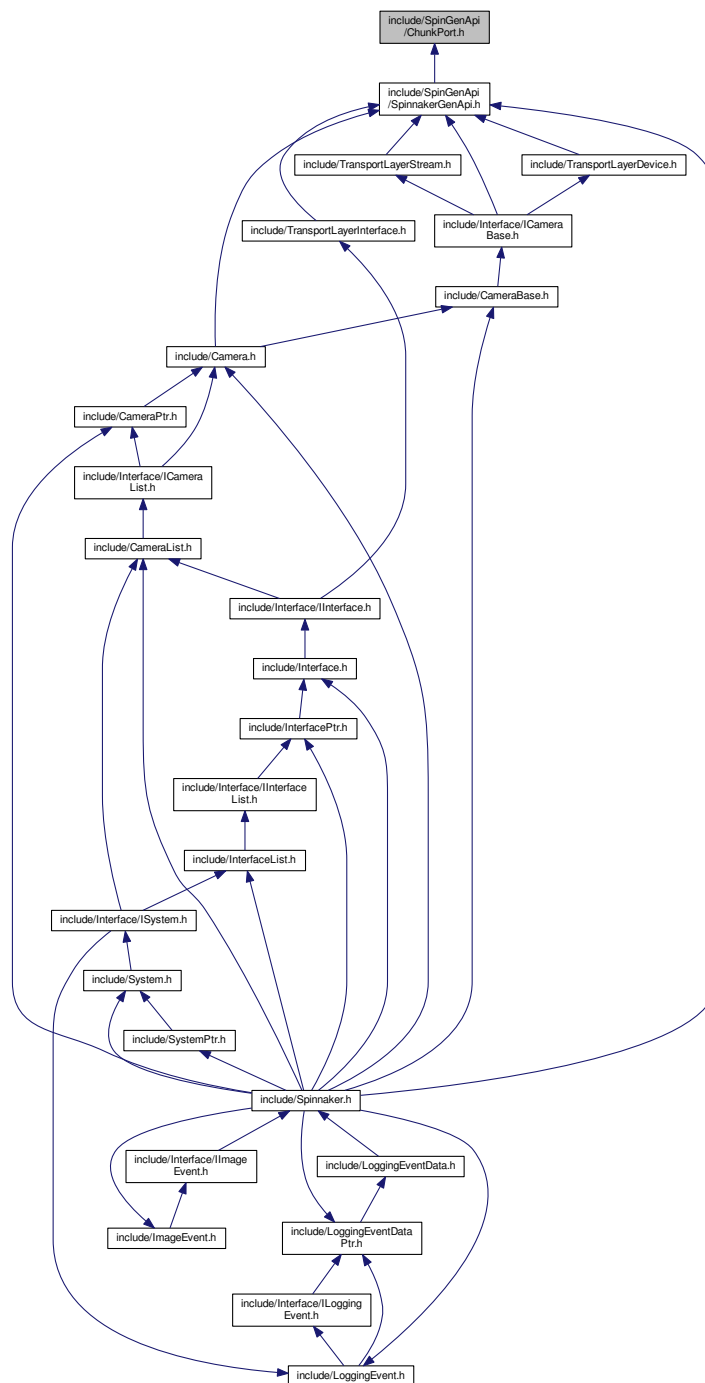
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

### 11.50 include/SpinGenApi/ChunkPort.h File Reference

Include dependency graph for ChunkPort.h:



This graph shows which files directly or indirectly include this file:



## Classes

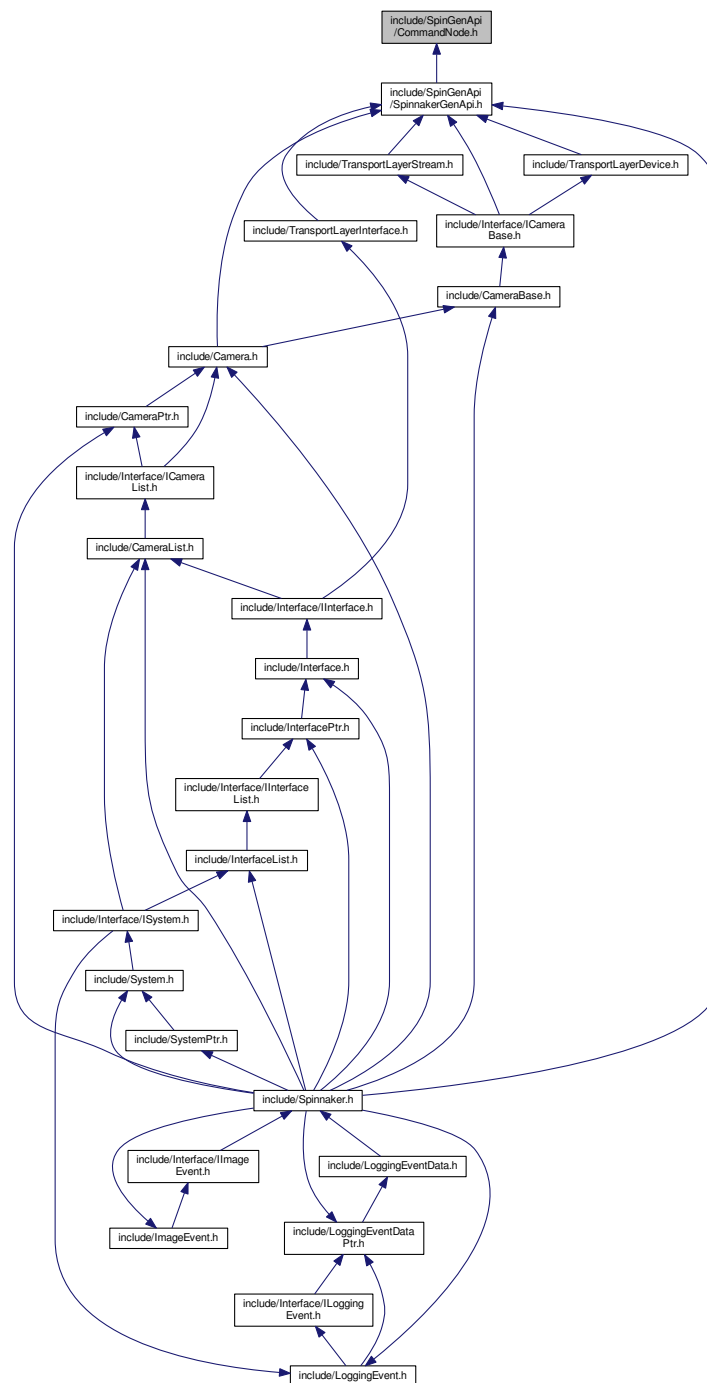
- class [CChunkPort](#)

*Port attachable to a chunk in a buffer.*





This graph shows which files directly or indirectly include this file:



## Classes

- class [CommandNode](#)  
*Interface for string properties.*

## Namespaces

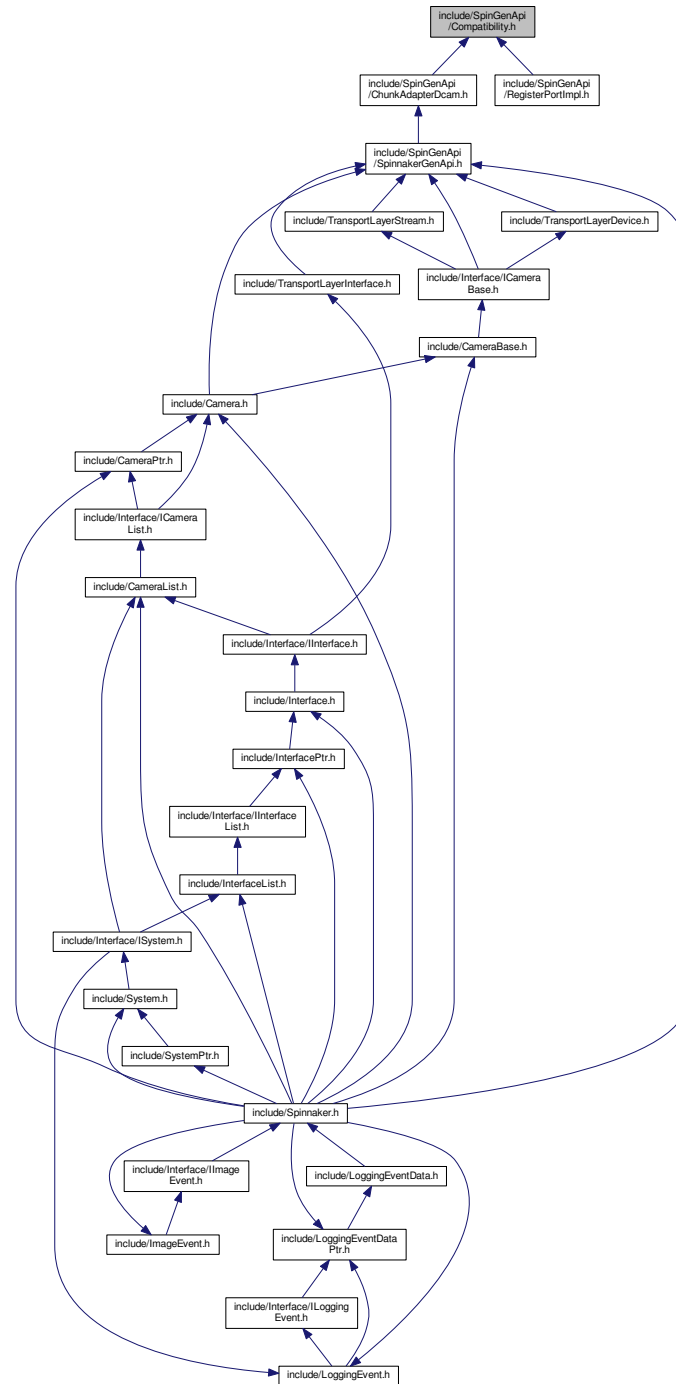
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- typedef CommandNode [CCommandRef](#)

## 11.52 include/SpinGenApi/Compatibility.h File Reference

This graph shows which files directly or indirectly include this file:



### Macros

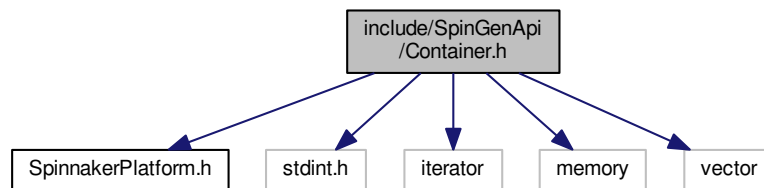
- `#define FMT_I64 "ll"`

### 11.52.1 Macro Definition Documentation

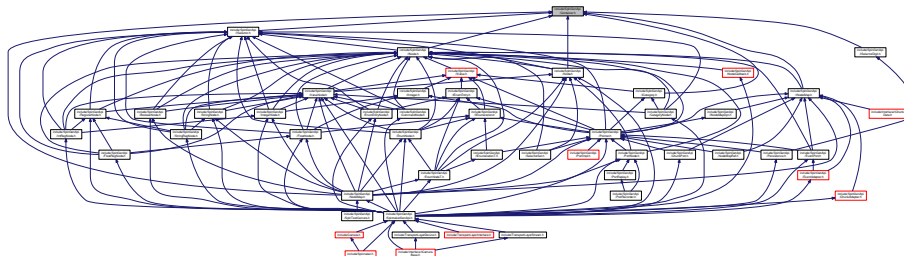
#### 11.52.1.1 `#define FMT_I64 "l"`

## 11.53 `include/SpinGenApi/Container.h` File Reference

Include dependency graph for `Container.h`:



This graph shows which files directly or indirectly include this file:



## 11.54 `include/SpinGenApi/Counter.h` File Reference

### Classes

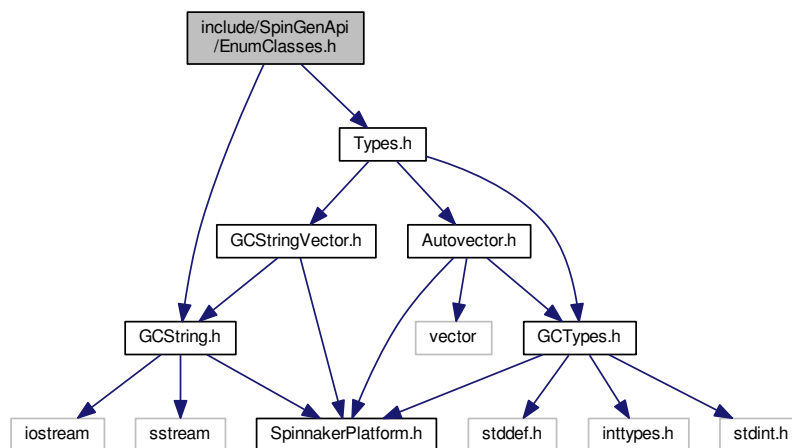
- class [Counter](#)  
Definition of a simple [Counter](#) class.

### Namespaces

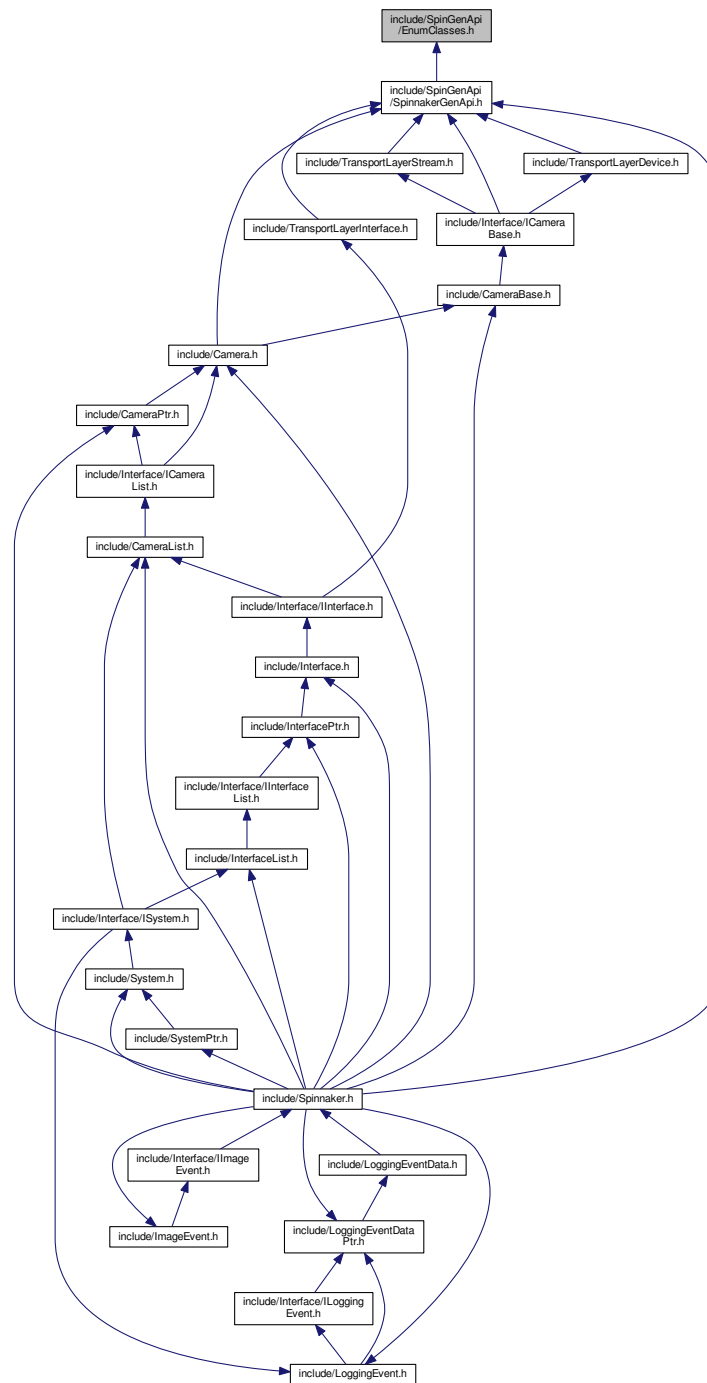
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.55 include/SpinGenApi/EnumClasses.h File Reference

Include dependency graph for EnumClasses.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ESignClass](#)  
Holds conversion methods for the sign enumeration.
- class [EEndianessClass](#)  
Holds conversion methods for the endianess enumeration.
- class [ERepresentationClass](#)

- class **EVisibilityClass**

- class `EAccessModeClass`

- class `ECachingModeClass`

- class `ENamespaceClass`

- class EYesNoClass

- class `EStandardNameSpaceClass`

- class ESlopeClass

- class `EDisplayNotationClass`

- class `EInputDirectionClass`

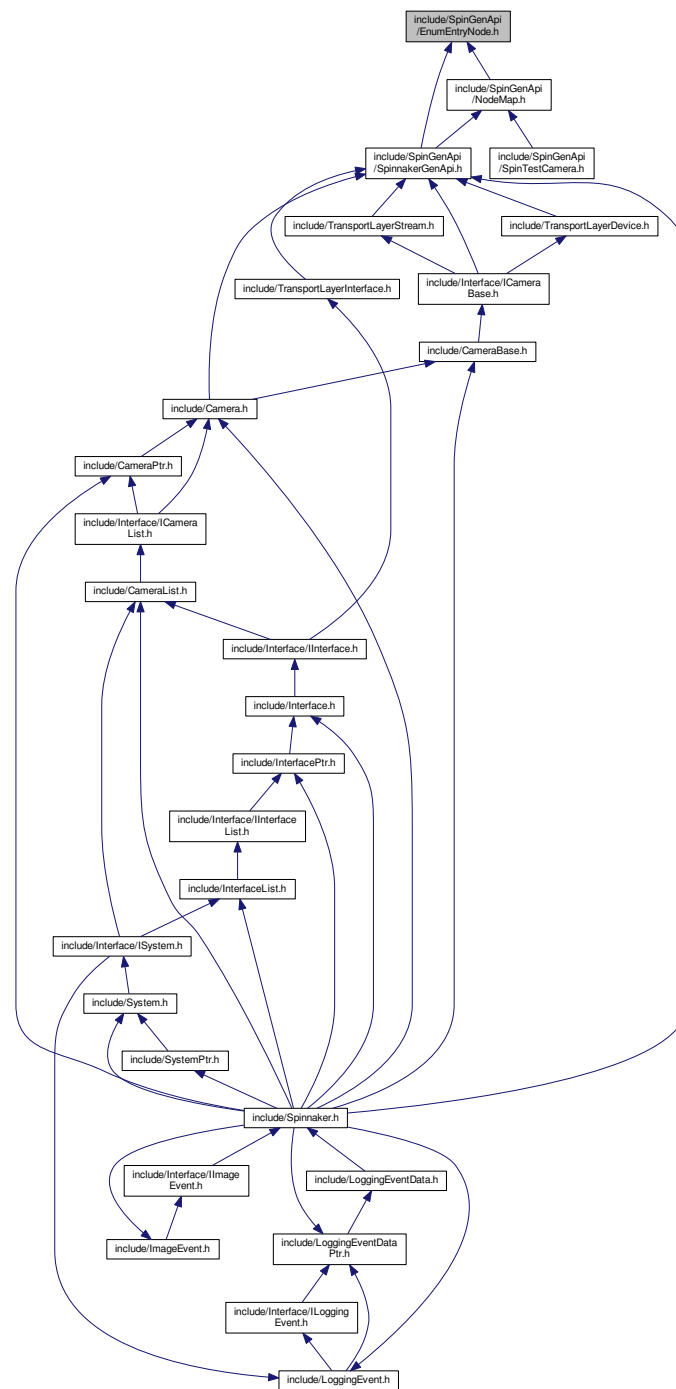
- class `EGenApiSchemaVersionClass`

## Namespaces

- Spinnaker
- Spinnaker::GenApi

## 11.56 include/SpinGenApi/EnumEntryNode.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

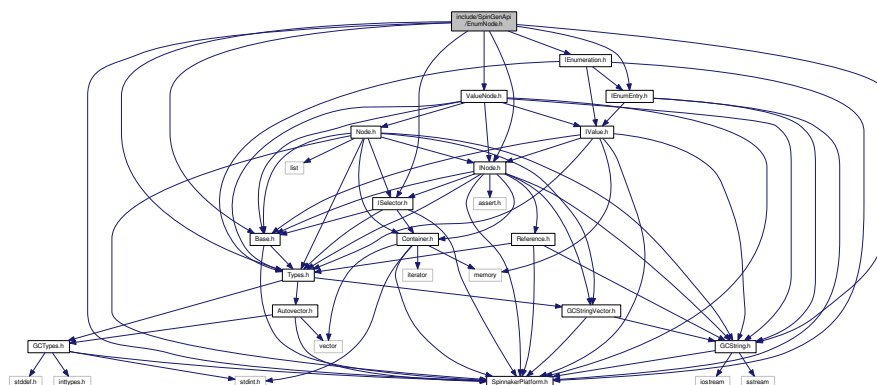
- class [EnumEntryNode](#)  
*Interface for string properties.*



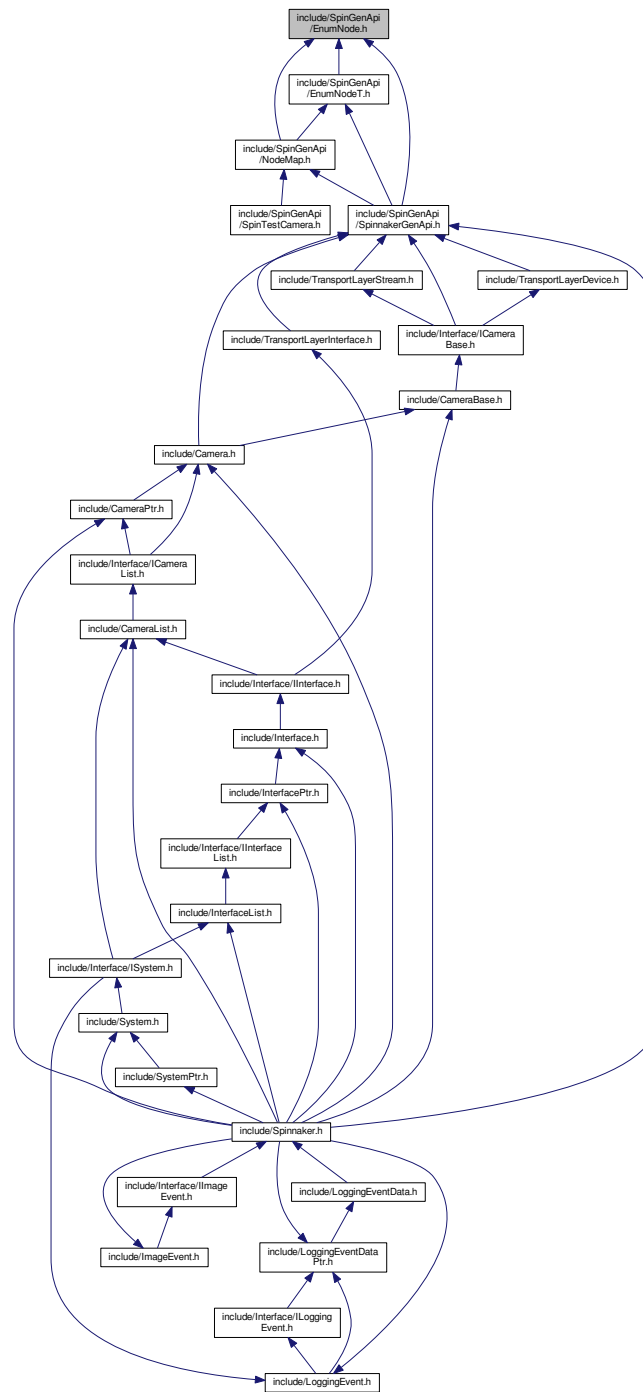
- Spinnaker
- Spinnaker::GenApi

- typedef EnumEntryNode CEnumEntryRef

Include dependency graph for EnumNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [EnumNode](#)  
*Interface for string properties.*

## Namespaces

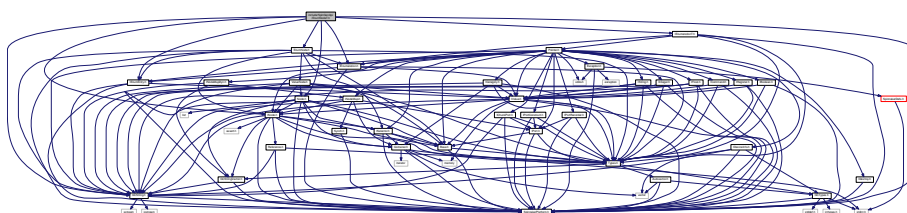
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

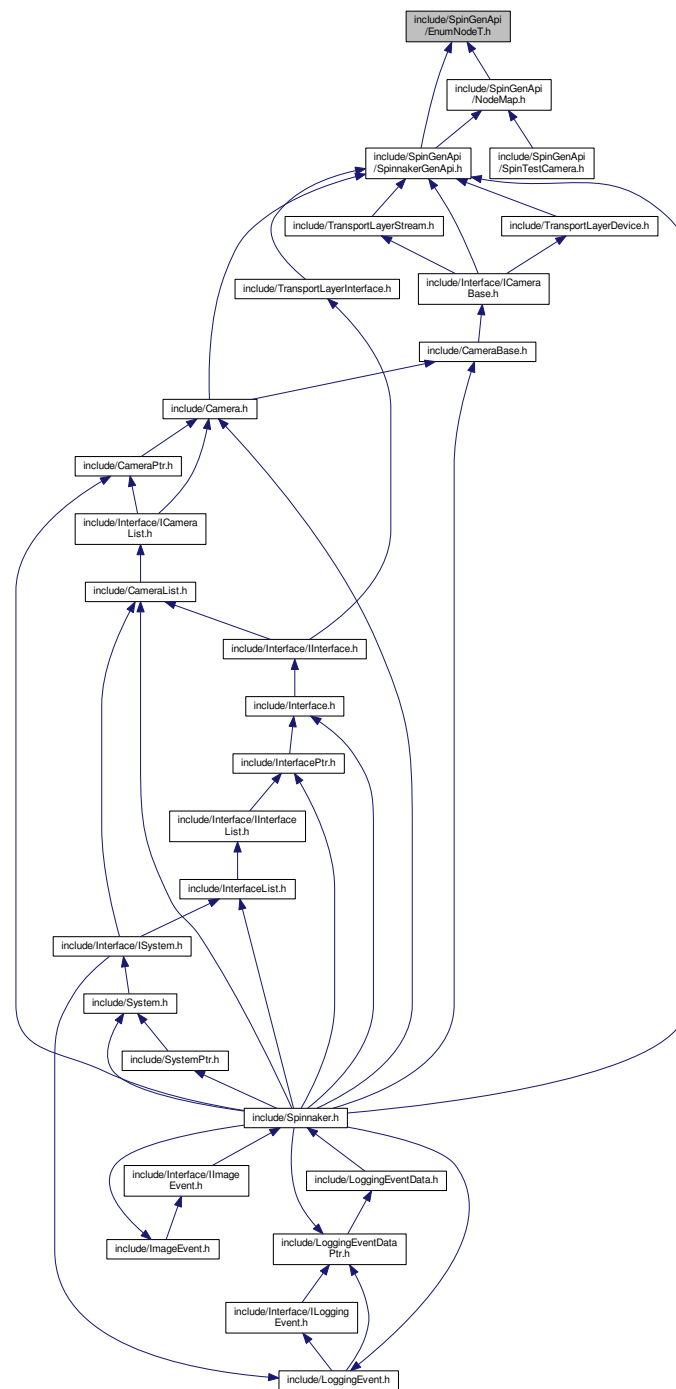
- typedef EnumNode [CEnumerationRef](#)

## 11.58 include/SpinGenApi/EnumNodeT.h File Reference

Include dependency graph for EnumNodeT.h:



This graph shows which files directly or indirectly include this file:



## Classes

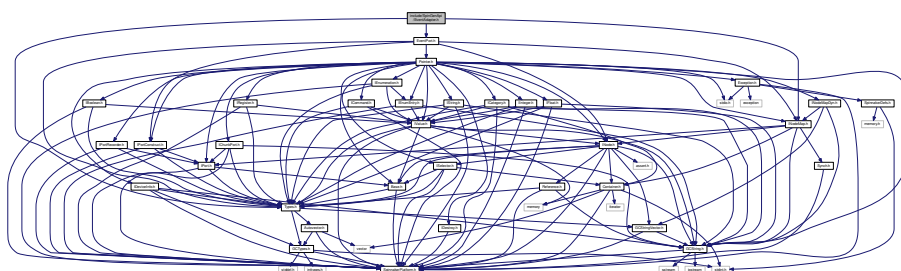
- class [CEnumerationTRef< EnumT >](#)  
*Interface* for string properties.

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.59 include/SpinGenApi/EventAdapter.h File Reference

Include dependency graph for EventAdapter.h:



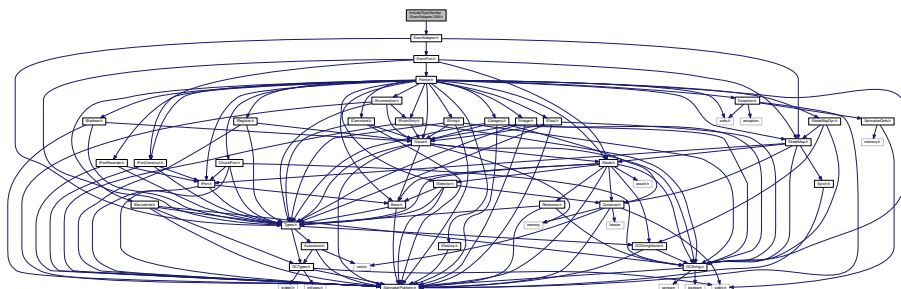


## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.60 include/SpinGenApi/EventAdapter1394.h File Reference

Include dependency graph for EventAdapter1394.h:





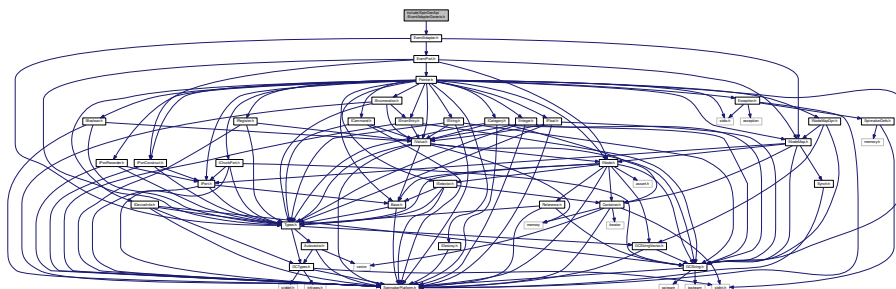


## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.61 include/SpinGenApi/EventAdapterGeneric.h File Reference

Include dependency graph for EventAdapterGeneric.h:



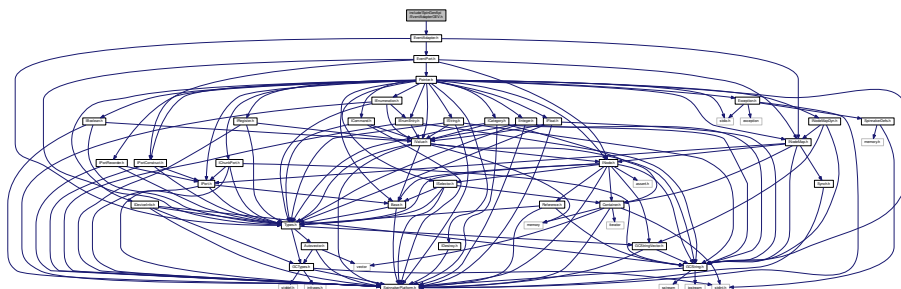


## Namespaces

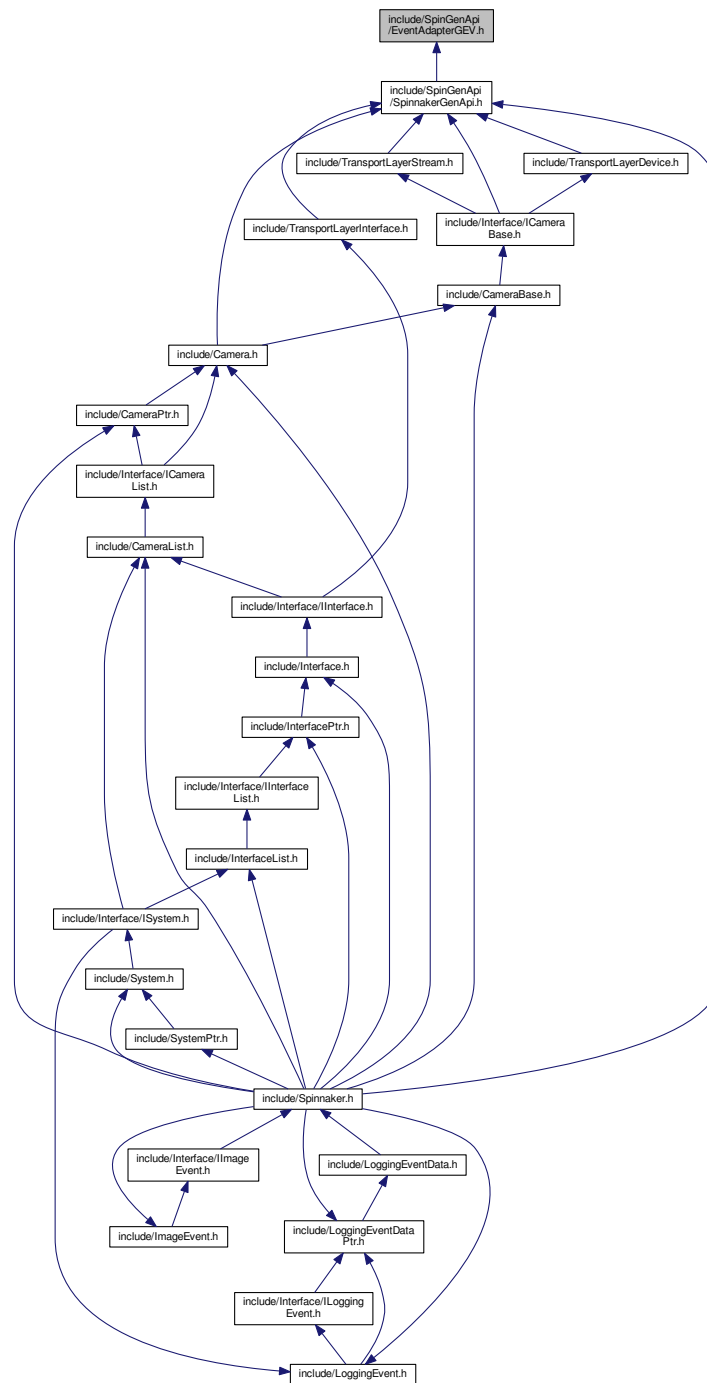
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.62 include/SpinGenApi/EventAdapterGEV.h File Reference

Include dependency graph for EventAdapterGEV.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [GVCP\\_REQUEST\\_HEADER](#)  
header of a GVCP request packet
- struct [GVCP\\_EVENT\\_ITEM\\_BASIC](#)  
layout of a GVCP event item (common to all types)
- struct [GVCP\\_EVENT\\_ITEM](#)

- layout of a GVCP event item (Extended ID flag not set)*

  - struct [GVCP\\_EVENT\\_REQUEST](#)

*Layout of a GVCP event request packet (Extended ID flag not set)*

  - struct [GVCP\\_EVENTDATA\\_REQUEST](#)

*Layout of a GVCP event data request packet (Extended ID flag not set)*

  - struct [GVCP\\_EVENT\\_ITEM\\_EXTENDED\\_ID](#)

*layout of a GVCP event item (Extended ID flag set)*

  - struct [GVCP\\_EVENT\\_REQUEST\\_EXTENDED\\_ID](#)

*Layout of a GVCP event request packet (Extended ID flag set)*

  - struct [GVCP\\_EVENTDATA\\_REQUEST\\_EXTENDED\\_ID](#)

*Layout of a GVCP event data request packet (Extended ID flag set)*

  - class [CEventAdapterGEV](#)

*Connects a GigE [Event](#) to a node map.*

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Enumerations

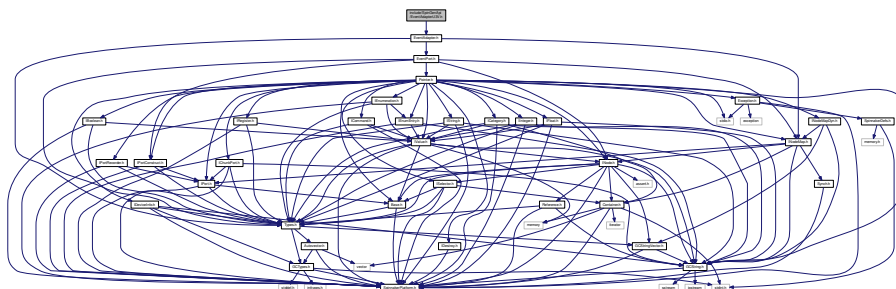
- enum [GVCP\\_MESSAGE\\_TAGS](#) {  
[TAG\\_EVENT\\_CMD](#) = 0xc0,  
[TAG\\_EVENTDATA\\_CMD](#) = 0xc2 }

## Variables

- const uint8\_t [COMMAND\\_MAGIC](#) = 0x42

## 11.63 include/SpinGenApi/EventAdapterU3V.h File Reference

Include dependency graph for EventAdapterU3V.h:





Entire event data message (without the variable-sized data field)

- class [CEventAdapterU3V](#)

Connects a U3V [Event](#) to a node map.

## Namespaces

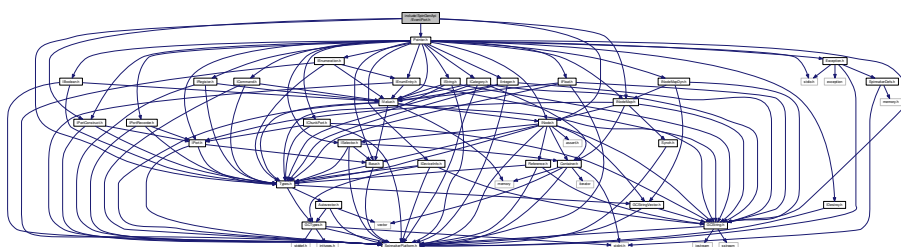
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Variables

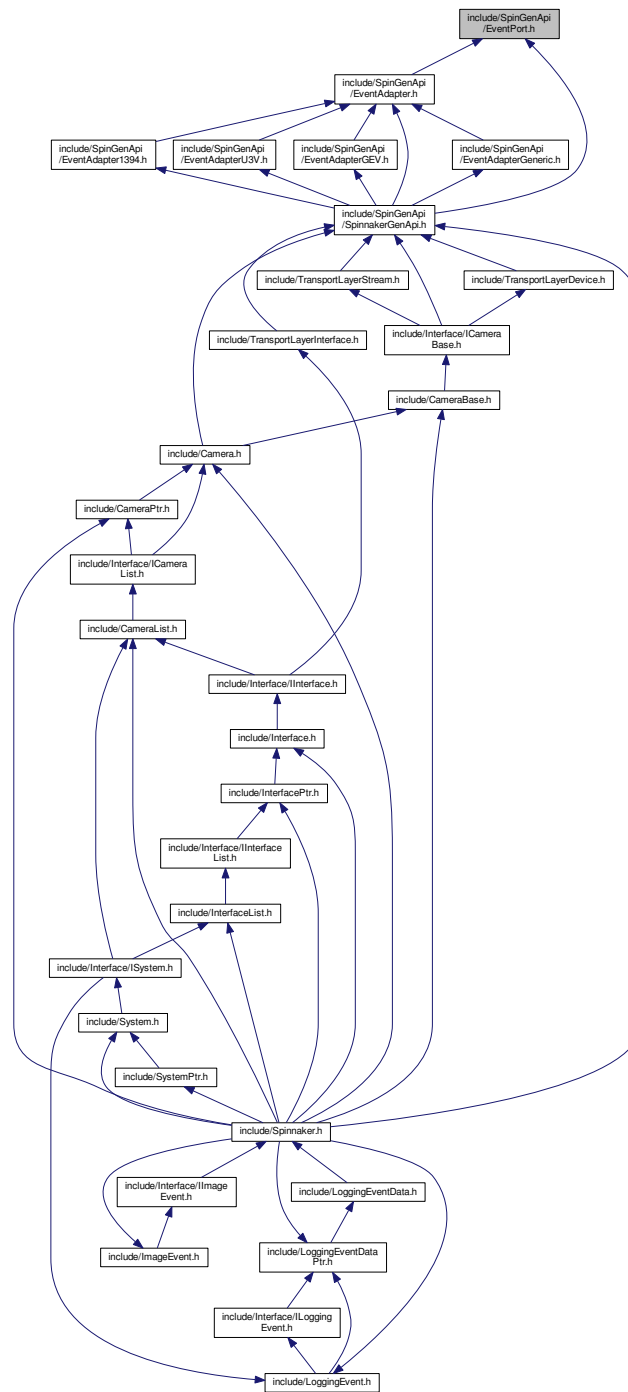
- const uint32\_t [U3V\\_EVENT\\_PREFIX](#) = 0x45563355
- const uint16\_t [GENCP\\_EVENT\\_CMD\\_ID](#) = 0x0C00
- const size\_t [GENCP\\_COMMAND\\_HEADER\\_SIZE](#) = sizeof([U3V\\_COMMAND\\_HEADER](#))
- const size\_t [GENCP\\_EVENT\\_BASIC\\_SIZE](#) = sizeof([U3V\\_EVENT\\_MESSAGE](#))

## 11.64 include/SpinGenApi/EventPort.h File Reference

Include dependency graph for EventPort.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [CEventPort](#)

*Port attachable to an event.*

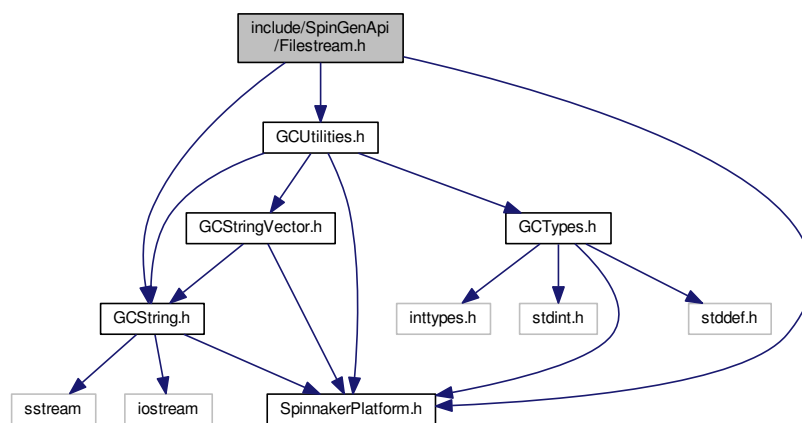


## Namespaces

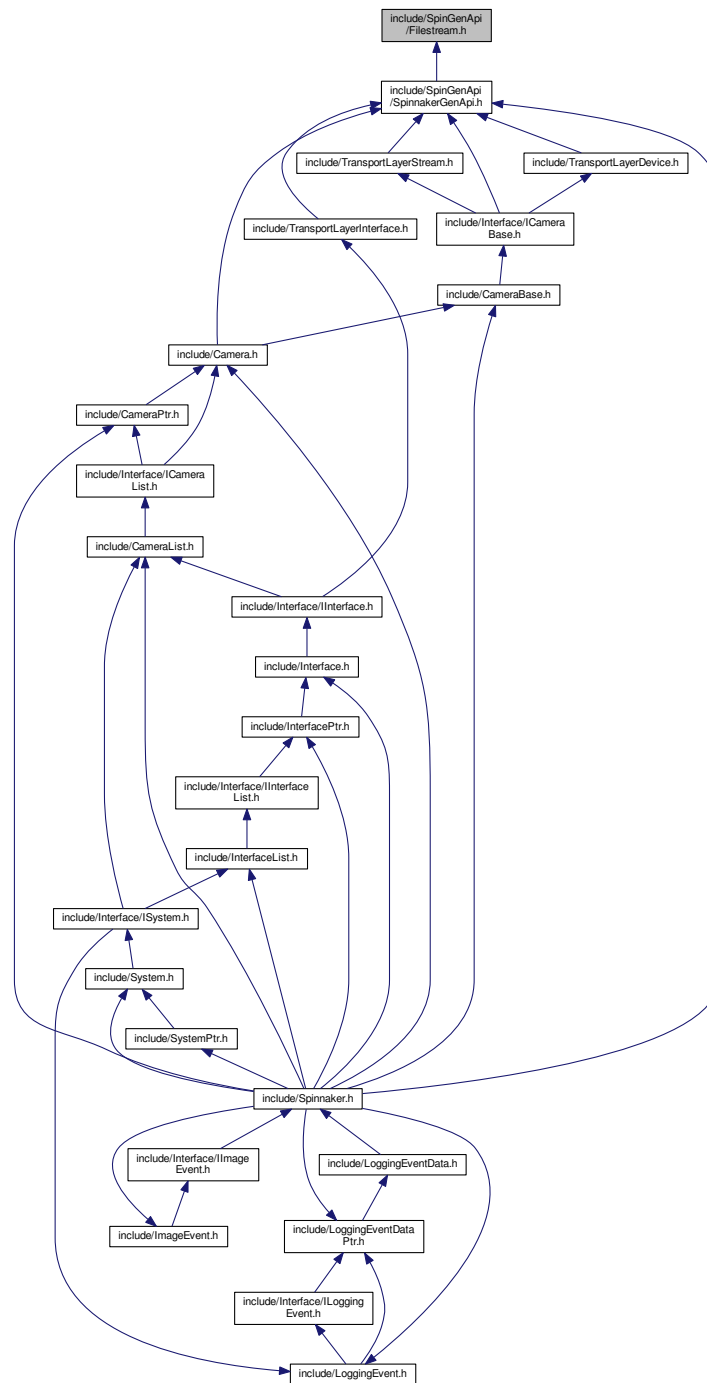
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.65 include/SpinGenApi/Filestream.h File Reference

Include dependency graph for FileStream.h:



This graph shows which files directly or indirectly include this file:



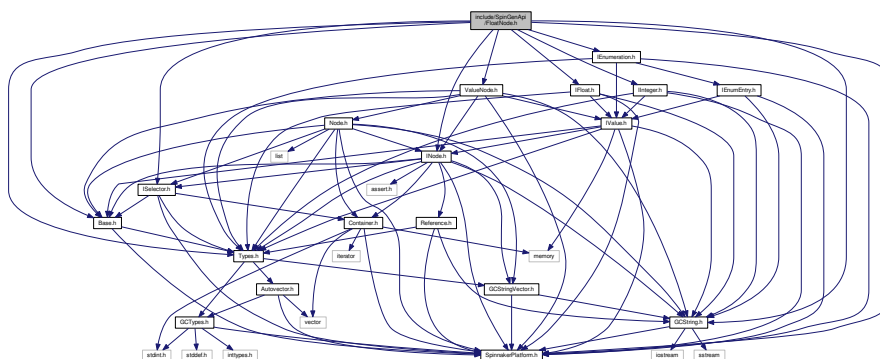
## Classes

- class [FileProtocolAdapter](#)  
*Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.*
- class [IDevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBase< CharType, Traits >](#)
- class [IDevFileStreamBase< CharType, Traits >](#)

- Spinnaker
- Spinnaker::GenApi

- typedef ODevFileStreamBase< char, std::char\_traits< char > > ODevFileStream
- typedef IDevFileStreamBase< char, std::char\_traits< char > > IDevFileStream

Include dependency graph for FloatNode.h:





## Namespaces

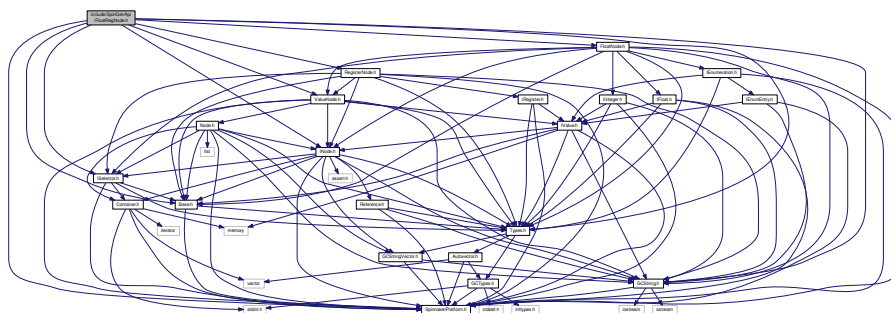
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- typedef FloatNode [CFloatRef](#)

## 11.67 include/SpinGenApi/FloatRegNode.h File Reference

Include dependency graph for FloatRegNode.h:



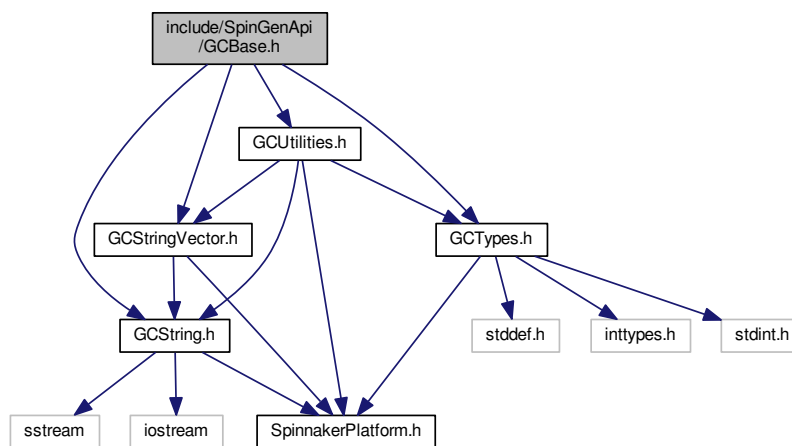


## Namespaces

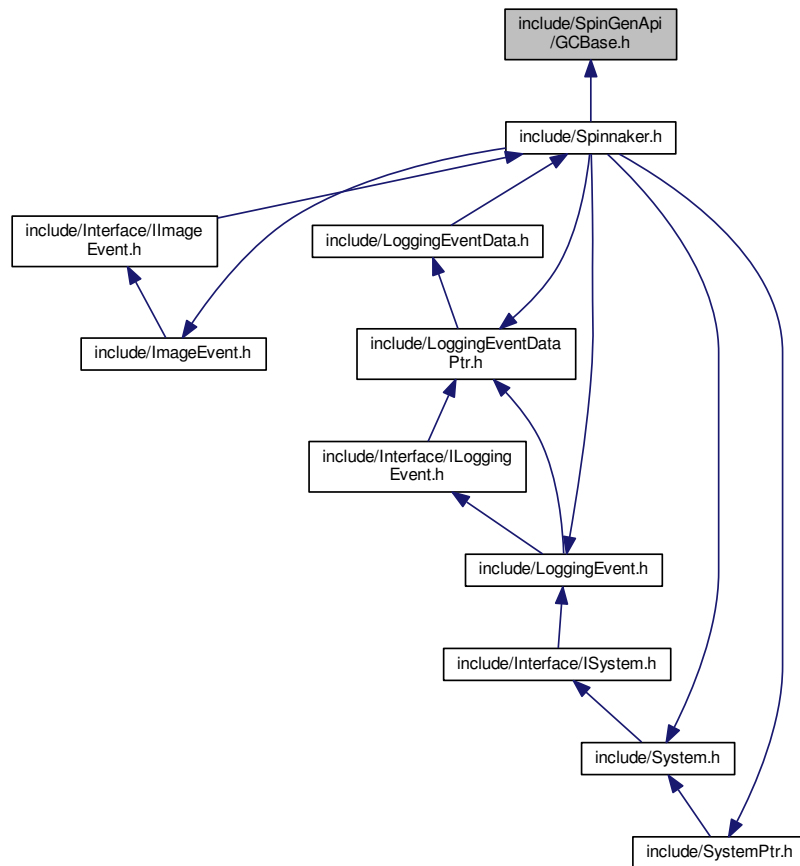
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.68 include/SpinGenApi/GCBase.h File Reference

Include dependency graph for GCBase.h:

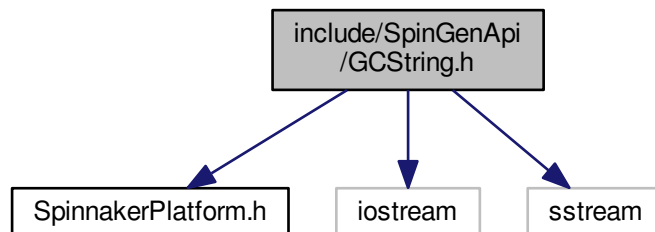


This graph shows which files directly or indirectly include this file:



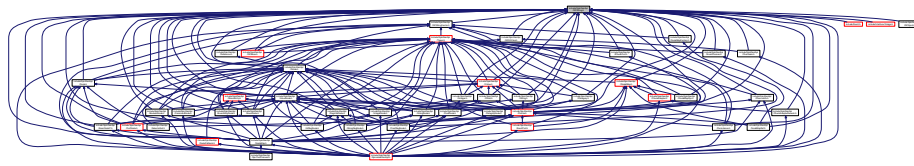
## 11.69 include/SpinGenApi/GCString.h File Reference

Include dependency graph for `GCString.h`:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gcstring](#)

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

## Macros

- `#define` [GCSTRING\\_NPOS](#) `size_t(-1)`

## Functions

- [SPINNAKER\\_API](#) `void` [ThrowBadAlloc](#) ()
- `std::istream &` [getline](#) (`std::istream &is`, [Spinnaker::GenICam::gcstring](#) &str)  
*STL getline.*
- `std::istream &` [getline](#) (`std::istream &is`, [Spinnaker::GenICam::gcstring](#) &str, `char` delim)  
*STL getline.*
- `std::ostream &` [operator<<](#) (`std::ostream &ostr`, `const` [Spinnaker::GenICam::gcstring](#) &str)  
*STL operator out.*
- `std::istream &` [operator>>](#) (`std::istream &istr`, [Spinnaker::GenICam::gcstring](#) &str)  
*STL operator in.*

### 11.69.1 Macro Definition Documentation

11.69.1.1 `#define` [GCSTRING\\_NPOS](#) `size_t(-1)`

### 11.69.2 Function Documentation

11.69.2.1 `std::ostream&` [operator<<](#) ( `std::ostream & ostr`, `const` [Spinnaker::GenICam::gcstring](#) & *str* ) `[inline]`

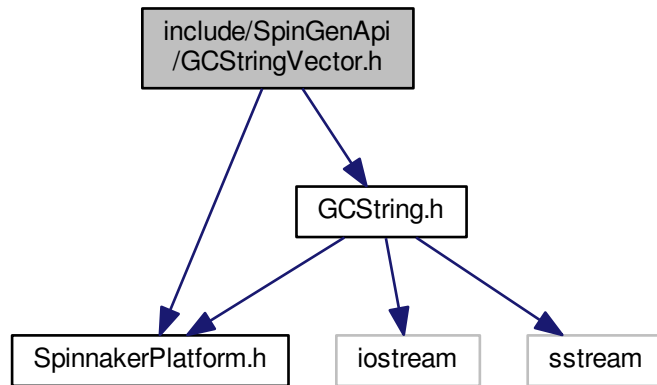
STL operator out.

11.69.2.2 `std::istream&` [operator>>](#) ( `std::istream & istr`, [Spinnaker::GenICam::gcstring](#) & *str* ) `[inline]`

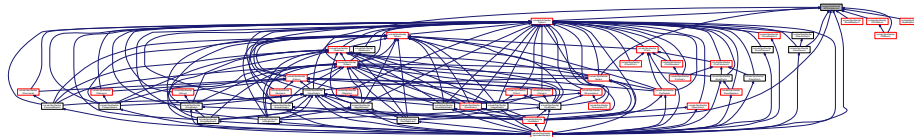
STL operator in.

## 11.70 include/SpinGenApi/GCStringVector.h File Reference

Include dependency graph for GCStringVector.h:

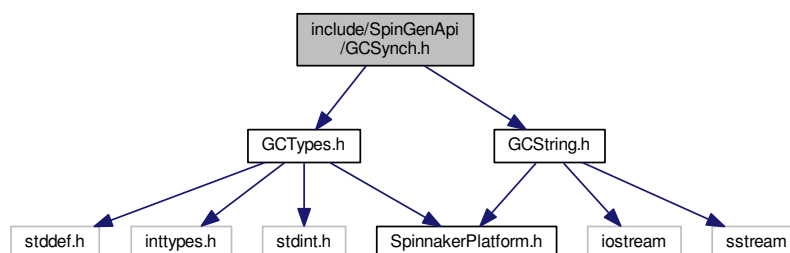


This graph shows which files directly or indirectly include this file:



## 11.71 include/SpinGenApi/GCSynch.h File Reference

Include dependency graph for GCSynch.h:



## Classes

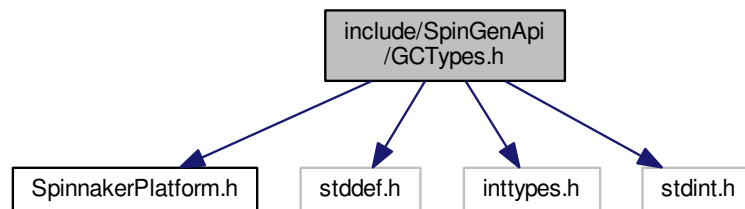
- class [CLock](#)  
*A lock class.*
- class [CLockEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)
- class [LockableObject< Object >](#)  
*Instance-Lock for an object.*
- class [LockableObject< Object >::Lock](#)  
*A scopelevel [Lock](#) class.*
- class [CGlobalLock](#)  
*Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)  
*Unlocks the global lock object on destruction.*

## Namespaces

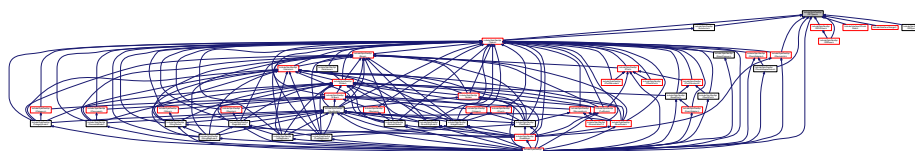
- [Spinnaker](#)
- [Spinnaker::GenICam](#)

## 11.72 include/SpinGenApi/GCTypes.h File Reference

Include dependency graph for GCTypes.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [Version\\_t](#)  
*Version.*

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

## Macros

- `#define __STDC_LIMIT_MACROS`
- `#define __STDC_CONSTANT_MACROS`
- `#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value */`
- `#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */`
- `#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */`
- `#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */`
- `#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */`
- `#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */`
- `#define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /* maximum signed int8 value */`
- `#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffffff80LL) /* minimum signed int8 value */`
- `#define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /* maximum unsigned int8 value */`

## Typedefs

- typedef float [float32\\_t](#)  
*32 bit floating point*
- typedef double [float64\\_t](#)  
*64 bit floating point*

### 11.72.1 Macro Definition Documentation

11.72.1.1 `#define __STDC_CONSTANT_MACROS`

11.72.1.2 `#define __STDC_LIMIT_MACROS`

11.72.1.3 `#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */`

11.72.1.4 `#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */`

11.72.1.5 `#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value */`

11.72.1.6 `#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */`

11.72.1.7 `#define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /* maximum signed int8 value */`

11.72.1.8 `#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffffff80LL) /* minimum signed int8 value */`

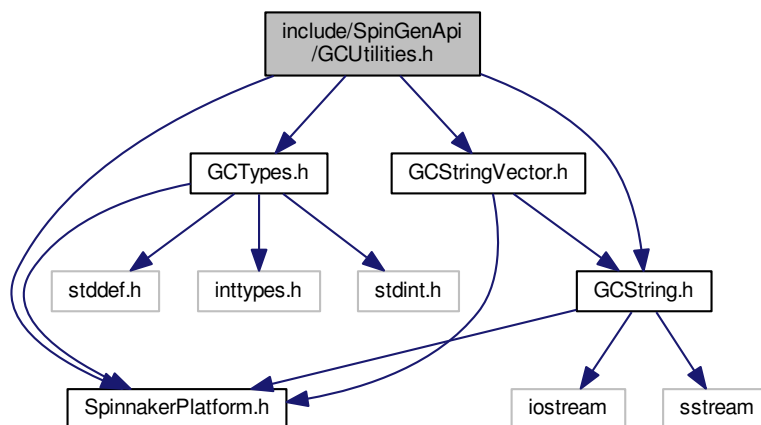
11.72.1.9 `#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */`

11.72.1.10 `#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */`

11.72.1.11 `#define GC_UINT8_MAX static_cast<uint8_t>(0x00000000000000ffULL) /* maximum unsigned int8 value */`

## 11.73 include/SpinGenApi/GCUtilities.h File Reference

Include dependency graph for GCUtilities.h:





- `#define USE_TEMP_CACHE_FILE 1`
- `#define GC_COUNTOF(arr) (sizeof (arr) / sizeof (arr)[0] )`
- `#define GENICAM_UNUSED(unused_var) ((void)(unused_var))`
- `#define GENICAM_DEPRECATED(FUNCTION) FUNCTION`
- `#define _TO_STRING(__stN) #__stN`
- `#define EXPAND_TO_STRINGISE(__stN) _TO_STRING( __stN )`
- `#define __LINE_STR__ EXPAND_TO_STRINGISE(__LINE__)`
- `#define __LOCATION__ __FILE__ "(" __LINE_STR__ ")"`
- `#define __OUTPUT_FORMATER__( _type) __LOCATION__ " : " _type " : "`
- `#define __WARN__ __OUTPUT_FORMATER__("WARNING")`
- `#define __ERR__ __OUTPUT_FORMATER__("ERROR")`
- `#define __TODO__ __OUTPUT_FORMATER__("TBD")`

## Functions

- `template<typename Td , typename Ts >`  
`Td INTEGRAL_CAST2 (Ts s)`  
*This verifies at runtime if there was no loss of data if an type Ts (e.g.*
- `template<typename T >`  
`T INTEGRAL_CAST (int64_t ll)`  
*This verifies at runtime if there was no loss of data if an int64\_t was downcast to type T (e.g.*
- `SPINNAKER_API bool DoesEnvironmentVariableExist (const Spinnaker::GenICam::gcstring &VariableName)`  
*Returns true if an environment variable exists.*
- `SPINNAKER_API gcstring GetValueOfEnvironmentVariable (const gcstring &VariableName)`  
*Retrieve the value of an environment variable.*
- `SPINNAKER_API bool GetValueOfEnvironmentVariable (const gcstring &VariableName, gcstring &VariableContent)`  
*Retrieve the value of an environment variable.*
- `SPINNAKER_API gcstring UrlEncode (const gcstring &Input)`  
*Converts \ to / and replaces all unsafe characters by their xx equivalent.*
- `SPINNAKER_API gcstring UrlDecode (const gcstring &Input)`  
*Replaces xx escapes by their char equivalent.*
- `SPINNAKER_API void ReplaceEnvironmentVariables (gcstring &Buffer, bool ReplaceBlankBy20=false)`  
*Replaces in a string and replace ' ' with %20.*
- `SPINNAKER_API gcstring GetGenICamCacheFolder (void)`  
*Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling SetGenICamCacheFolder().*
- `SPINNAKER_API gcstring GetGenICamLogConfig (void)`  
*Retrieve the path of the GenICam logging properties file.*
- `SPINNAKER_API gcstring GetGenICamCLProtocolFolder (void)`  
*Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling SetGenICamCLProtocolFolder().*
- `SPINNAKER_API void SetGenICamCacheFolder (const gcstring &path)`  
*Stores the path of the GenICam cache folder.*
- `SPINNAKER_API void SetGenICamLogConfig (const gcstring &path)`  
*Stores the path of the GenICam logging properties file.*
- `SPINNAKER_API void SetGenICamCLProtocolFolder (const gcstring &path)`  
*Stores the path of the CLProtocol folder.*
- `SPINNAKER_API void Tokenize (const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")`  
*splits str input string into a list of tokens using the delimiter*

- [SPINNAKER\\_API](#) void [GetFiles](#) (const gcstring &FileTemplate, gcstring\_vector &FileNames, const bool DirectoriesOnly=false)

*Gets a list of files or directories matching a given FileTemplate.*

- [SPINNAKER\\_API](#) gcstring [GetModulePathFromFunction](#) (void \*pFunction)

*Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.*

### 11.73.1 Macro Definition Documentation

11.73.1.1 `#define __ERR__ __OUTPUT_FORMATER__("ERROR")`

11.73.1.2 `#define __LINE_STR__ EXPAND_TO_STRINGISE(__LINE__)`

11.73.1.3 `#define __LOCATION__ __FILE__ "(" __LINE_STR__ ")"`

11.73.1.4 `#define __OUTPUT_FORMATER__( _type ) __LOCATION__ " : " _type " : "`

11.73.1.5 `#define __TODO__ __OUTPUT_FORMATER__("TBD")`

11.73.1.6 `#define __WARN__ __OUTPUT_FORMATER__("WARNING")`

11.73.1.7 `#define _TO_STRING( __stN ) #__stN`

11.73.1.8 `#define EXPAND_TO_STRINGISE( __stN ) _TO_STRING( __stN )`

11.73.1.9 `#define GC_COUNTOF( arr ) (sizeof (arr) / sizeof (arr)[0] )`

11.73.1.10 `#define GENICAM_DEPRECATED( FUNCTION ) FUNCTION`

11.73.1.11 `#define GENICAM_UNUSED( unused_var ) ((void)(unused_var))`

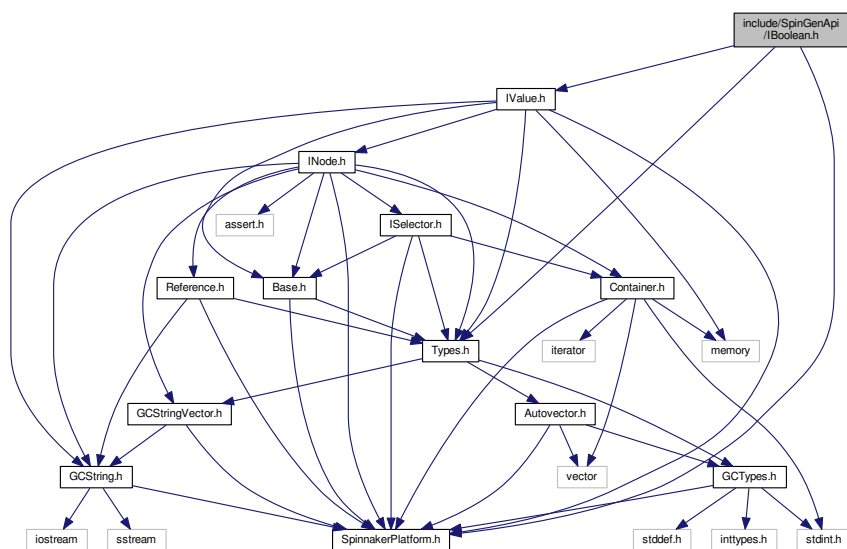
11.73.1.12 `#define USE_TEMP_CACHE_FILE 1`

11.73.1.13 `#define USE_TEMP_CACHE_FILE 1`



## 11.74 include/SpinGenApi/IBoolean.h File Reference

Include dependency graph for IBoolean.h:





## Variables

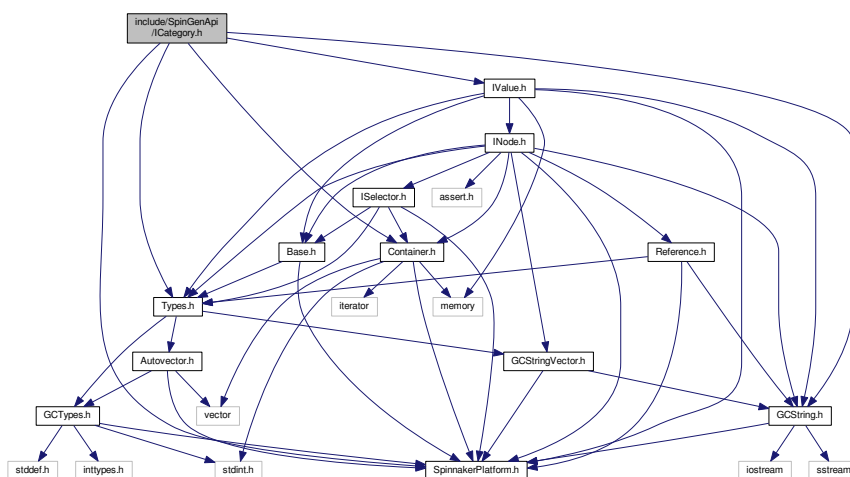
- [interface SPINNAKER\\_API\\_ABSTRACT IBoolean](#)

*Interface for Boolean properties.*

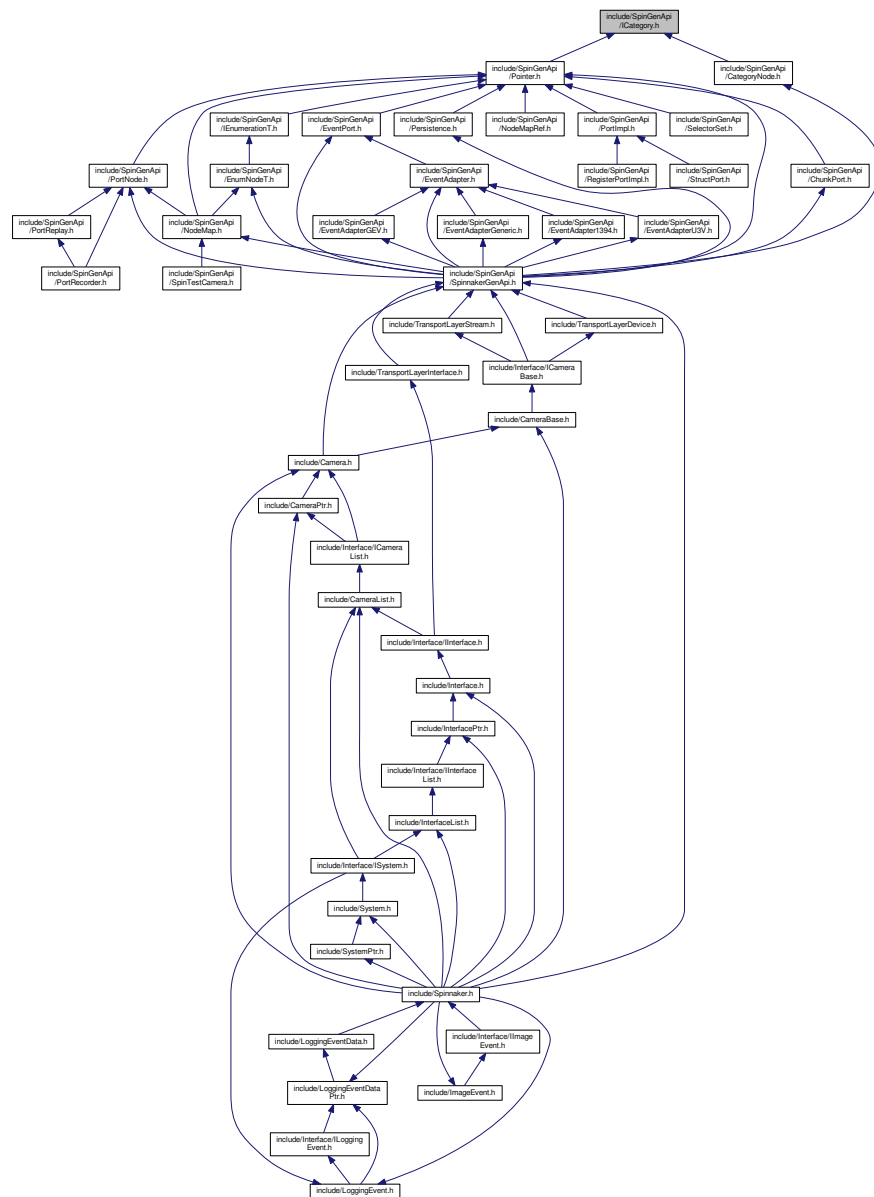
- [interface SPINNAKER\\_API\\_ABSTRACT](#) bool [Verify](#) = true) = 0

## 11.75 include/SpinGenApi/ICategory.h File Reference

Include dependency graph for ICategory.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

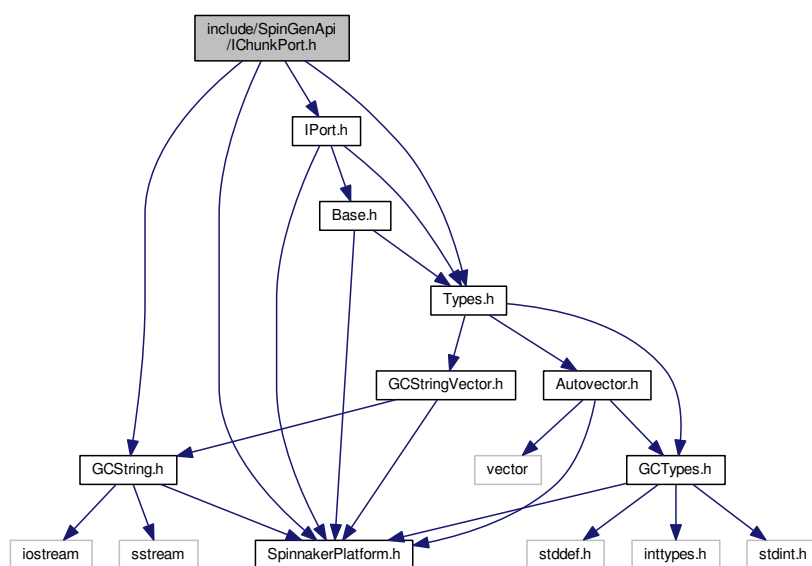
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT ICategory](#)  
*Gives access to a category node.*

## 11.76 include/SpinGenApi/IChunkPort.h File Reference

Include dependency graph for IChunkPort.h:





## Functions

- virtual EYesNo [CacheChunkData](#) () const =0

*Indicates if the chunk a adapter must hold a cached version of the chunk data.*

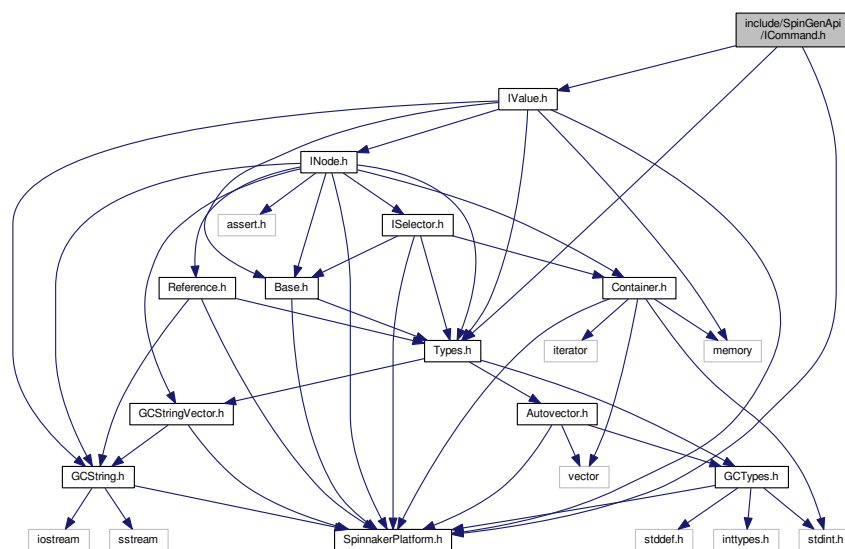
## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IChunkPort](#)

*Interface for ports attached to a chunk.*

## 11.77 include/SpinGenApi/ICommand.h File Reference

Include dependency graph for ICommand.h:







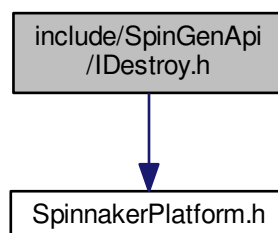
## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT ICommand](#)

*Interface for command like properties.*

## 11.78 include/SpinGenApi/IDestroy.h File Reference

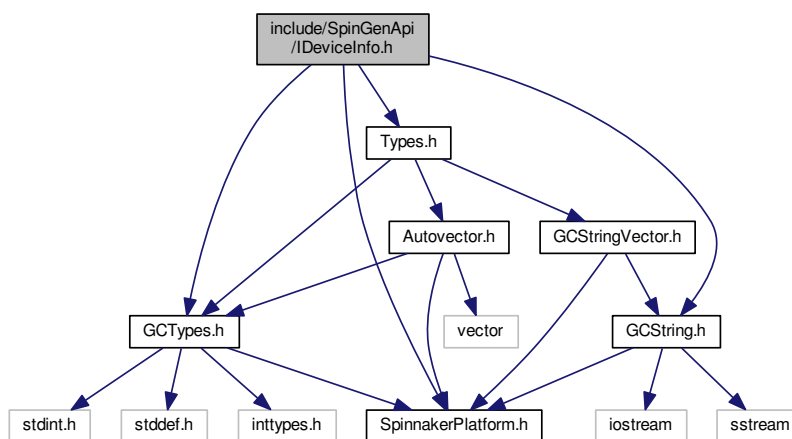
Include dependency graph for IDestroy.h:





## 11.79 include/SpinGenApi/IDeviceInfo.h File Reference

Include dependency graph for IDeviceInfo.h:





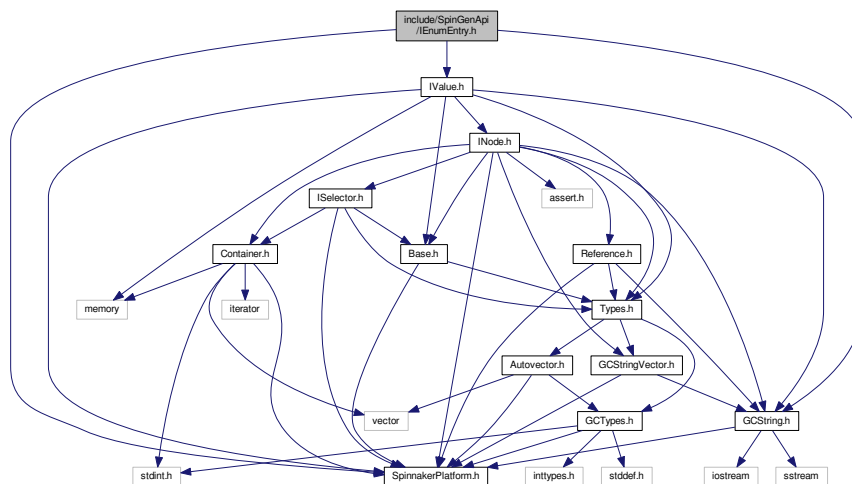
- virtual GenICam::gcstring [GetStandardNameSpace](#) ()=0  
*Get the standard name space.*
- virtual void [GetGenApiVersion](#) (GenICam::Version\_t &Version, uint16\_t &Build)=0  
*Get the version of the DLL's [GenApi](#) implementation.*
- virtual void [GetSchemaVersion](#) (GenICam::Version\_t &Version)=0  
*Get the schema version number.*
- virtual void [GetDeviceVersion](#) (GenICam::Version\_t &Version)=0  
*Get the version of the device description file.*
- virtual GenICam::gcstring [GetProductGuid](#) ()=0  
*Get the Guid describing the product.*
- virtual GenICam::gcstring [GetVersionGuid](#) ()=0  
*Get the Guid describing the product version.*

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IDeviceInfo](#)  
*Interface to get information about the device (= nodemap)*

## 11.80 include/SpinGenApi/IEnumEntry.h File Reference

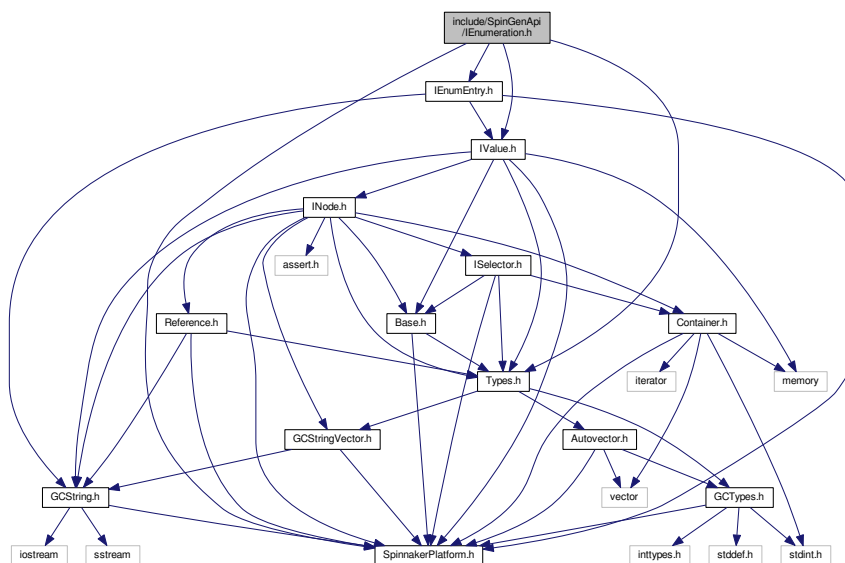
Include dependency graph for IEnumEntry.h:



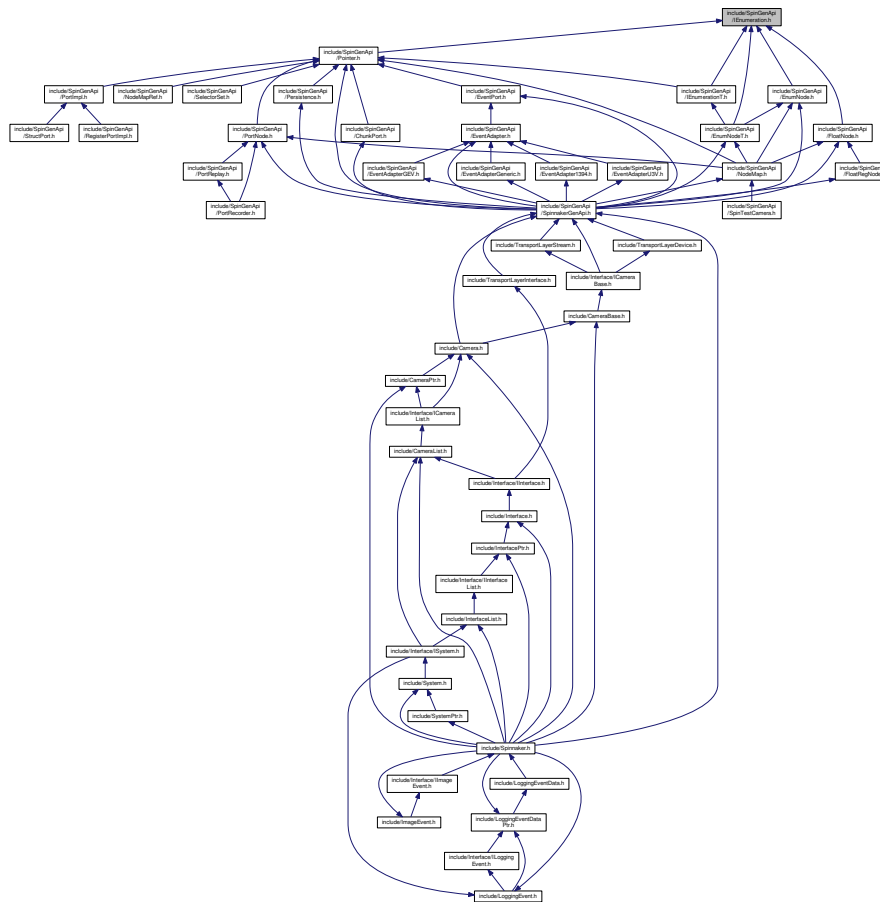


## 11.81 include/SpinGenApi/IEnumeration.h File Reference

Include dependency graph for IEnumeration.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

- virtual void [GetEntries](#) (NodeList\_t &Entries)=0  
*Get list of entry nodes.*
- virtual IEnumeration & [operator=](#) (const GenICam::gcstring &ValueStr)=0  
*Set string node value.*
- virtual void [SetIntValue](#) (int64\_t Value, bool Verify=true)=0  
*Set integer node value.*
- virtual GenICam::gcstring [operator\\*](#) ()=0  
*Get string node value.*
- virtual int64\_t [GetIntValue](#) (bool Verify=false, bool IgnoreCache=false)=0  
*Get integer node value.*
- virtual IEnumEntry \* [GetEntryByName](#) (const GenICam::gcstring &Symbolic)=0  
*Get an entry node by name.*
- virtual IEnumEntry \* [GetEntry](#) (const int64\_t IntValue)=0  
*Get an entry node by its IntValue.*
- virtual IEnumEntry \* [GetCurrentEntry](#) (bool Verify=false, bool IgnoreCache=false)=0  
*Get the current entry.*



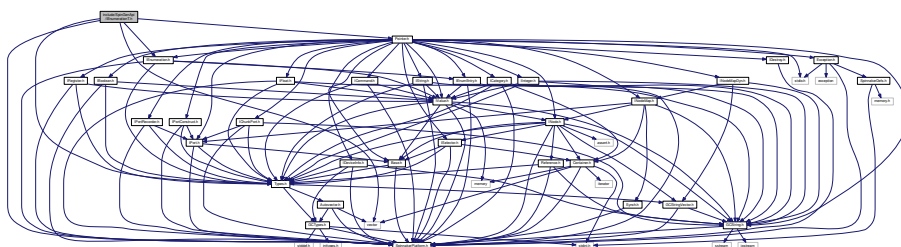
## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IEnumeration](#)

*Interface* for enumeration properties.

## 11.82 include/SpinGenApi/IEnumerationT.h File Reference

Include dependency graph for IEnumerationT.h:





*Set node value.*

- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0

*Get node value.*

- virtual bool [operator\(\)](#) () const

*Get node value.*

- virtual IEnumeration & [operator=](#) (const GenICam::gcstring &ValueStr)=0

*Set string node value.*

- virtual IEnumEntry \* [GetEntry](#) (const EnumT Value)=0

*returns the EnumEntry object belonging to the Value*

- virtual IEnumEntry \* [GetCurrentEntry](#) (bool Verify=false, bool IgnoreCache=false)=0

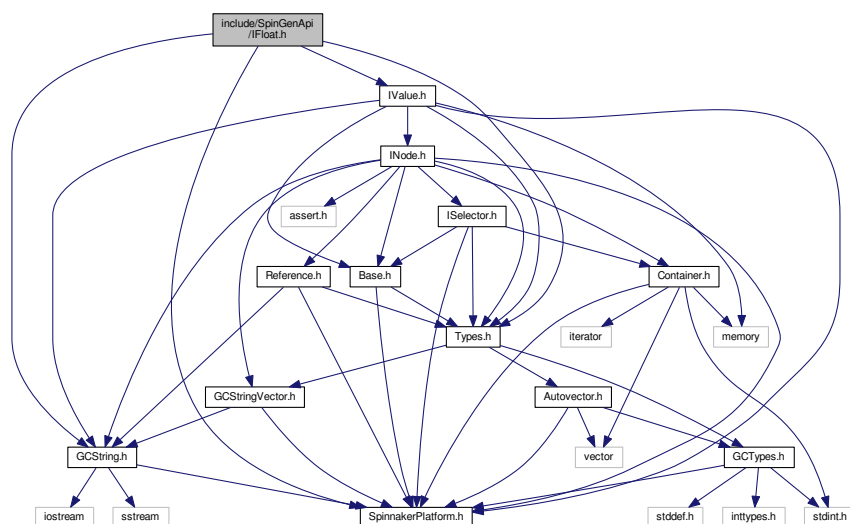
*Get the current entry.*

## Variables

- template<typename EnumT >  
[interface SPINNAKER\\_API\\_ABSTRACT IEnumerationT](#)  
*Interface for enumeration properties.*
- template<typename EnumT >  
[interface SPINNAKER\\_API\\_ABSTRACT](#) virtual public [IEnumReference](#)  
*Interface to construct an enum reference.*

## 11.83 include/SpinGenApi/IFloat.h File Reference

Include dependency graph for IFloat.h:





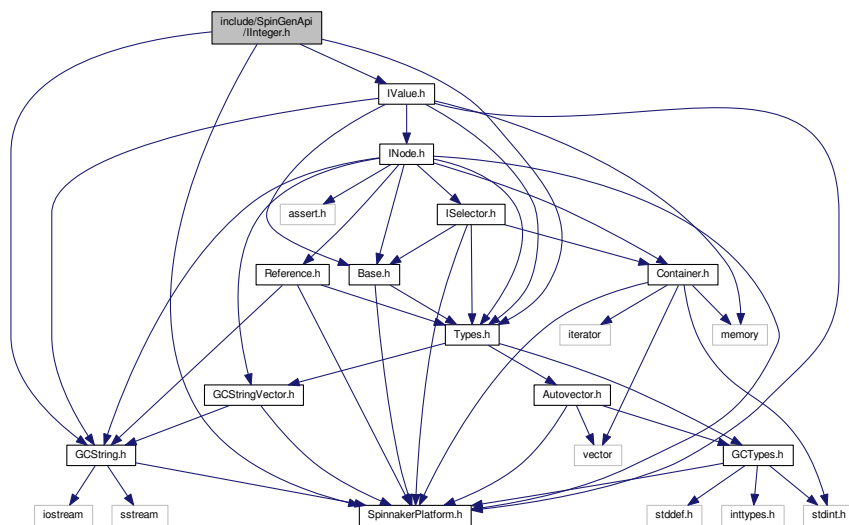
- Get maximum value allowed.*
  - virtual bool [HasInc](#) ()=0
  - True if the float has a constant increment.*
- virtual EIncMode [GetIncMode](#) ()=0
  - Get increment mode.*
- virtual double [GetInc](#) ()=0
  - Get the constant increment if there is any.*
- virtual double\_\_autovector\_\_t [GetListOfValidValues](#) (bool bounded=true)=0
  - Get list of valid value.*
- virtual ERepresentation [GetRepresentation](#) ()=0
  - Get recommended representation.*
- virtual GenICam::gcstring [GetUnit](#) () const =0
  - Get the physical unit name.*
- virtual EDisplayNotation [GetDisplayNotation](#) () const =0
  - Get the way the float should be converted to a string.*
- virtual int64\_\_t [GetDisplayPrecision](#) () const =0
  - Get the precision to be used when converting the float to a string.*
- virtual void [ImposeMin](#) (double Value)=0
  - Restrict minimum value.*
- virtual void [ImposeMax](#) (double Value)=0
  - Restrict maximum value.*

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IFloat](#)  
*Interface for float properties.*

## 11.84 include/SpinGenApi/IInteger.h File Reference

Include dependency graph for IInteger.h:





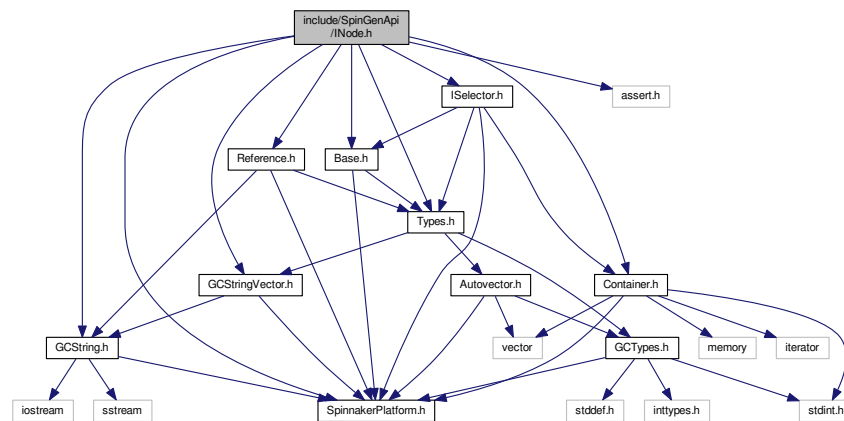
- Get maximum value allowed.*
- virtual EIncMode [GetIncMode](#) ()=0
- Get increment mode.*
- virtual double [GetInc](#) ()=0
- Get the constant increment if there is any.*
- virtual double\_\_autovector\_t [GetListOfValidValues](#) (bool bounded=true)=0
- Get list of valid value.*
- virtual ERepresentation [GetRepresentation](#) ()=0
- Get recommended representation.*
- virtual GenICam::gcstring [GetUnit](#) () const =0
- Get the physical unit name.*
- virtual void [ImposeMin](#) (int64\_t Value)=0
- Restrict minimum value.*
- virtual void [ImposeMax](#) (int64\_t Value)=0
- Restrict maximum value.*

## Variables

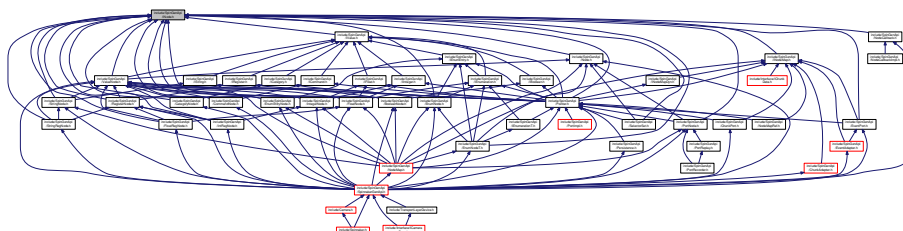
- [interface SPINNAKER\\_API\\_ABSTRACT Integer](#)  
*Interface for integer properties.*

## 11.85 include/SpinGenApi/INode.h File Reference

Include dependency graph for INode.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- typedef node\_vector [NodeList\\_t](#)  
*a list of node references*
- typedef intptr\_t [CallbackHandleType](#)  
*the callback handle for nodes*

## Functions

- virtual GenApi::ENamespace [GetNameSpace](#) () const =0  
*Get name space.*
- virtual EVisibility [GetVisibility](#) () const =0  
*Get the recommended visibility of the node.*
- virtual void [InvalidateNode](#) ()=0  
*Indicates that the node's value may have changed.*
- virtual bool [IsCacheable](#) () const =0  
*Is the node value cacheable.*
- virtual EYesNo [IsAccessModeCacheable](#) () const =0  
*True if the AccessMode can be cached.*
- virtual ECachingMode [GetCachingMode](#) () const =0  
*Get Caching Mode.*
- virtual int64\_t [GetPollingTime](#) () const =0  
*recommended polling time (for non-cacheable nodes)*
- virtual GenICam::gcstring [GetToolTip](#) ()=0  
*Get tool tip.*
- virtual GenICam::gcstring [GetDescription](#) () const =0  
*Get a long description of the node.*
- virtual GenICam::gcstring [GetDisplayName](#) () const =0  
*Get a name string for display.*
- virtual GenICam::gcstring [GetDeviceName](#) () const =0  
*Get a name of the device.*
- virtual void [GetChildren](#) (GenApi::NodeList\_t &Children, ELinkType LinkType=ctReadingChildren) const =0  
*Get all nodes this node directly depends on.*
- virtual void [GetParents](#) (GenApi::NodeList\_t &Parents) const =0  
*Gets all nodes this node is directly depending on.*
- virtual CallbackHandleType [RegisterCallback](#) (CNodeCallback \*pCallback)=0  
*Register change callback Takes ownership of the CNodeCallback object.*
- virtual bool [DeregisterCallback](#) (CallbackHandleType hCallback)=0  
*De register change callback Destroys CNodeCallback object.*
- virtual INodeMap \* [GetNodeMap](#) () const =0  
*Retrieves the central node map.*
- virtual GenICam::gcstring [GetEventID](#) () const =0  
*Get the EventId of the node.*
- virtual bool [IsStreamable](#) () const =0  
*True if the node is streamable.*



- virtual void [GetPropertyNames](#) (GenICam::gcstring\_vector &PropertyNames) const =0  
*Returns a list of the names all properties set during initialization.*
- virtual bool [GetProperty](#) (const GenICam::gcstring &PropertyName, GenICam::gcstring &ValueStr, GenICam::gcstring &AttributeStr)=0  
*Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.*
- virtual void [ImposeAccessMode](#) (EAccessMode ImposedAccessMode)=0  
*Imposes an access mode to the natural access mode of the node.*
- virtual void [ImposeVisibility](#) (EVisibility ImposedVisibility)=0  
*Imposes a visibility to the natural visibility of the node.*
- virtual INode \* [GetAlias](#) () const =0  
*Retrieves the a node which describes the same feature in a different way.*
- virtual INode \* [GetCastAlias](#) () const =0  
*Retrieves the a node which describes the same feature so that it can be casted.*
- virtual GenICam::gcstring [GetDocuURL](#) () const =0  
*Gets a URL pointing to the documentation of that feature.*
- virtual bool [IsDeprecated](#) () const =0  
*True if the node should not be used any more.*
- virtual EInterfaceType [GetPrincipalInterfaceType](#) () const =0  
*Get the type of the main interface of a node.*
- virtual bool [IsFeature](#) () const =0  
*True if the node can be reached via category nodes from a category node named "Root".*
- virtual bool [operator==](#) (int nullPtr) const =0
- virtual bool [operator!=](#) (int nullPtr) const =0
- bool [IsReadable](#) (EAccessMode AccessMode)  
*Tests if readable.*
- bool [IsReadable](#) (const IBase \*p)  
*Checks if a node is readable.*
- bool [IsReadable](#) (const IBase &r)  
*Checks if a node is readable.*
- bool [IsWritable](#) (EAccessMode AccessMode)  
*Tests if writable.*
- bool [IsWritable](#) (const IBase \*p)  
*Checks if a node is writable.*
- bool [IsWritable](#) (const IBase &r)  
*Checks if a node is writable.*
- bool [IsImplemented](#) (EAccessMode AccessMode)  
*Tests if implemented.*
- bool [IsImplemented](#) (const IBase \*p)  
*Checks if a node is implemented.*
- bool [IsImplemented](#) (const IBase &r)  
*Checks if a node is implemented.*
- bool [IsAvailable](#) (EAccessMode AccessMode)  
*Tests if available.*
- bool [IsAvailable](#) (const IBase \*p)  
*Checks if a node is available.*
- bool [IsAvailable](#) (const IBase &r)  
*Checks if a node is available.*
- EAccessMode [Combine](#) (EAccessMode Peter, EAccessMode Paul)  
*Computes which access mode the two guards allow together.*
- bool [IsVisible](#) (EVisibility Visibility, EVisibility MaxVisibility)

Tests Visibility CAVE : this relies on the EVisibility enum's coding.

- EVisibility [Combine](#) (EVisibility Peter, EVisibility Paul)

Computes which visibility the two guards allow together.

- bool [IsCacheable](#) (ECachingMode CachingMode)

Tests Cacheability.

- ECachingMode [Combine](#) (ECachingMode Peter, ECachingMode Paul)

Computes which CachingMode results from a combination.

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT INode](#)

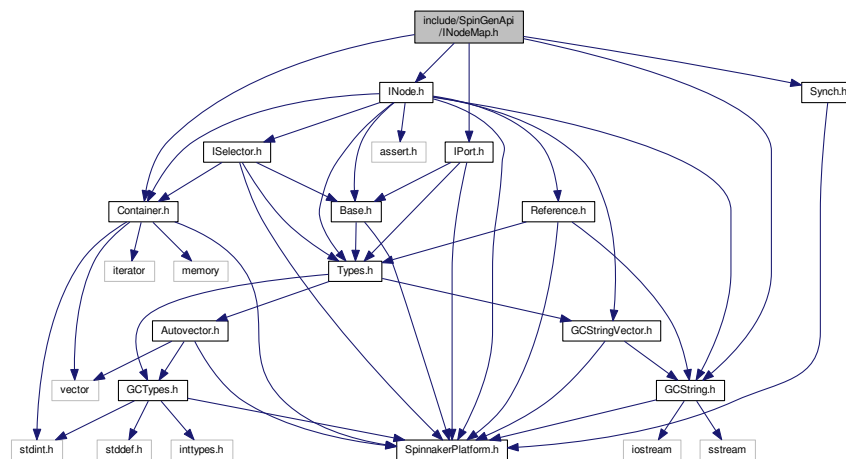
[Interface](#) common to all nodes.

- [interface SPINNAKER\\_API\\_ABSTRACT](#) virtual public [IReference](#)

[Interface](#) to construct a reference.

## 11.86 include/SpinGenApi/INodeMap.h File Reference

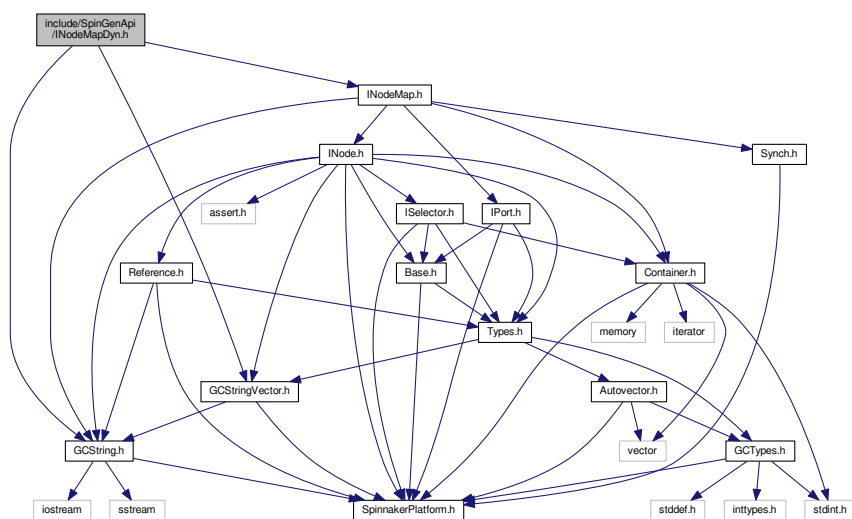
Include dependency graph for INodeMap.h:





## 11.87 include/SpinGenApi/INodeMapDyn.h File Reference

Include dependency graph for INodeMapDyn.h:



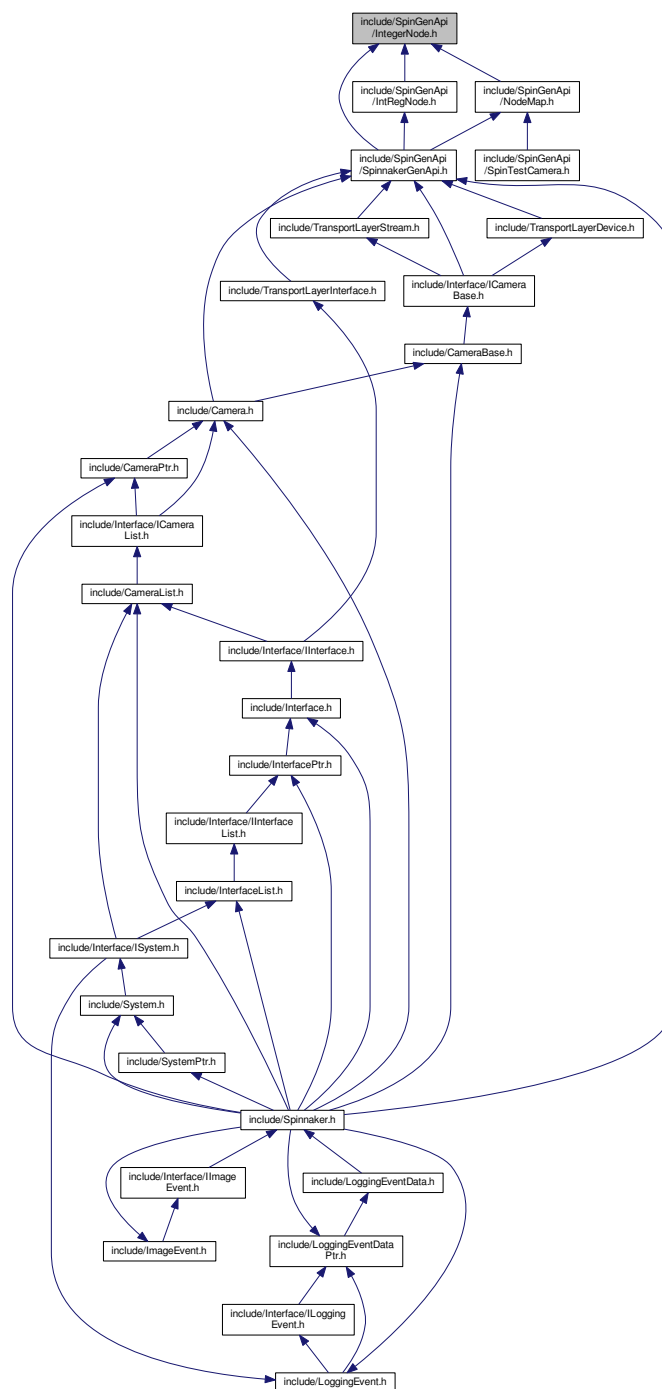
[illegible]

- Spinnaker
- Spinnaker::GenApi

- virtual void [LoadXMLFromFile](#) (const GenICam::gcstring &FileName)=0  
*Loads an XML from a file.*
- virtual void [LoadXMLFromFileInject](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)=0  
*Loads an XML from a file with injection.*
- virtual void [LoadXMLFromString](#) (const GenICam::gcstring &XMLData)=0  
*Loads an XML from a string.*
- virtual void [LoadXMLFromStringInject](#) (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)=0  
*Loads an XML from a string with injection.*



This graph shows which files directly or indirectly include this file:



## Classes

- class `IntegerNode`  
*Interface for string properties.*

## Namespaces

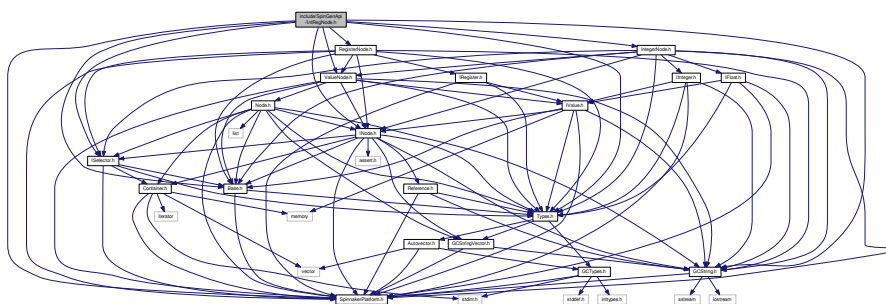
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- typedef IntegerNode [CIntegerRef](#)

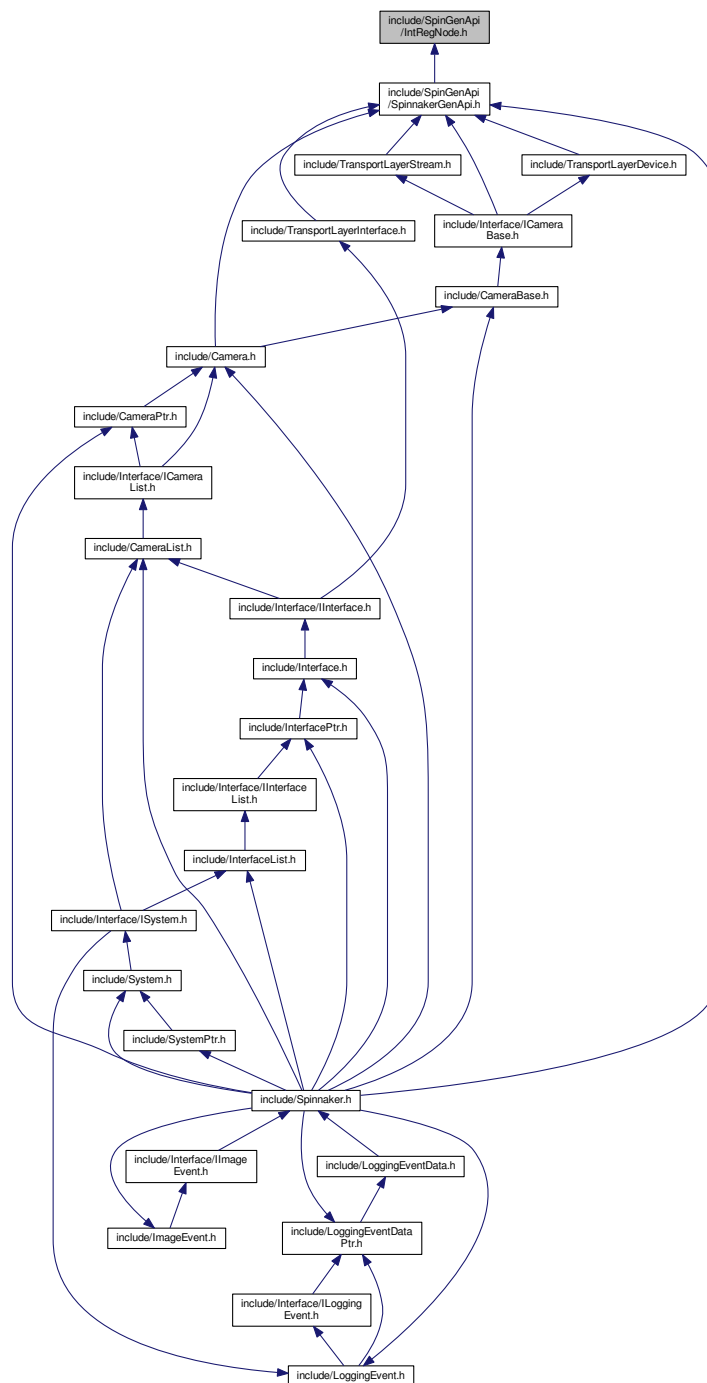
## 11.89 include/SpinGenApi/IntRegNode.h File Reference

Include dependency graph for IntRegNode.h:





This graph shows which files directly or indirectly include this file:



## Classes

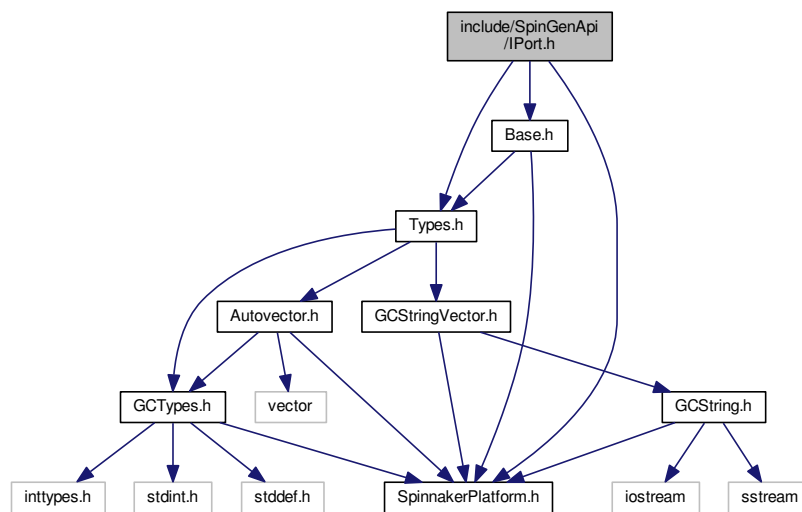
- class `IntRegNode`  
*Interface for string properties.*

## Namespaces

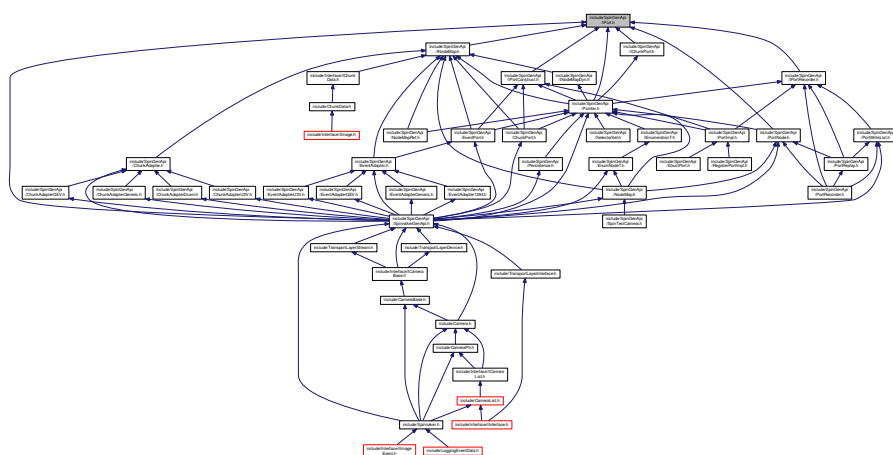
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.90 include/SpinGenApi/IPort.h File Reference

Include dependency graph for IPort.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

- virtual void [Write](#) (const void \*pBuffer, int64\_t Address, int64\_t Length)=0

*Writes a chunk of bytes to the port.*

## Variables

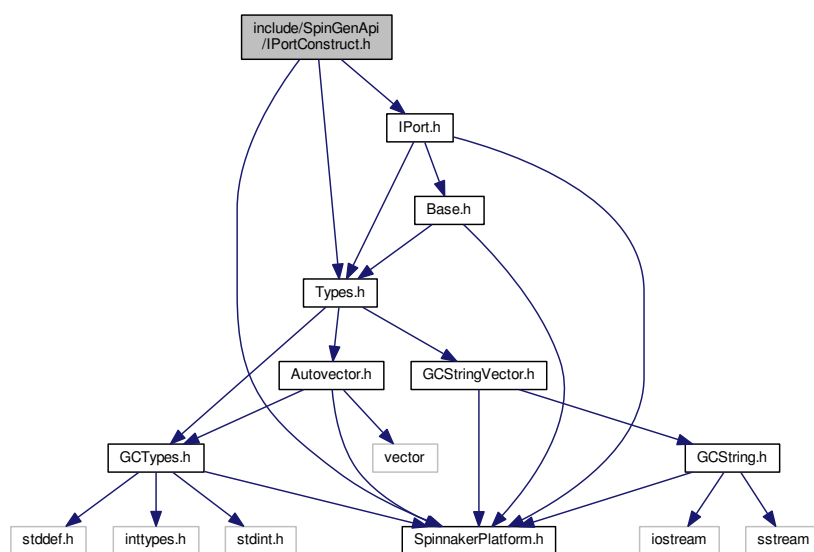
- [interface SPINNAKER\\_API\\_ABSTRACT IPort](#)

*Interface for ports.*

- [interface SPINNAKER\\_API\\_ABSTRACT int64\\_t Address](#)
- [interface SPINNAKER\\_API\\_ABSTRACT int64\\_t int64\\_t Length = 0](#)

## 11.91 include/SpinGenApi/IPortConstruct.h File Reference

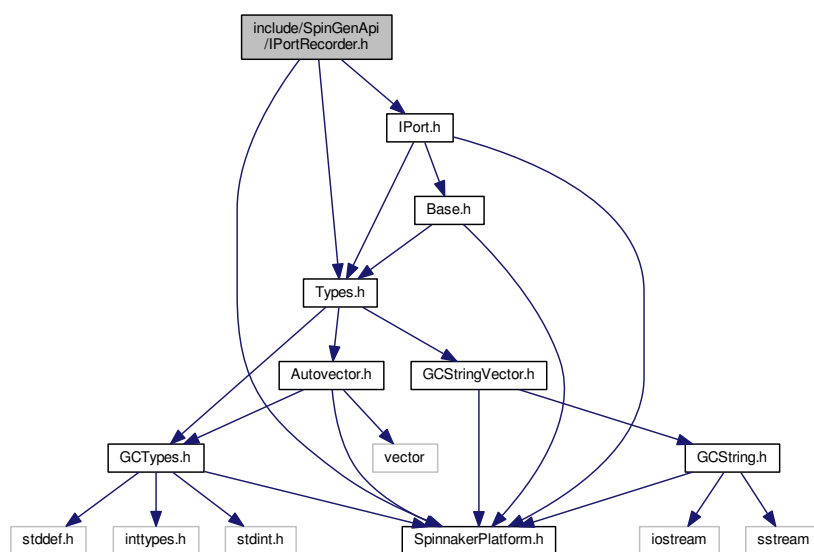
Include dependency graph for IPortConstruct.h:





## 11.92 include/SpinGenApi/IPortRecorder.h File Reference

Include dependency graph for IPortRecorder.h:



[illegible]

- Spinnaker
- Spinnaker::GenApi

- virtual void **Replay** (IPort \*pPort)=0  
*Replays the write command to the given port interface.*
- virtual void **SetCookie** (const int64\_t Value)=0  
*Sets a cookie in case the port implementation want to cache a command list.*
- virtual int64\_t **GetCookie** ()=0  
*Gets the cookie a port implementation may have set for caching a command list.*
- virtual void **StopRecording** ()=0  
*Stops recording.*

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IPortWriteList](#)
- [interface SPINNAKER\\_API\\_ABSTRACT IPortReplay](#)

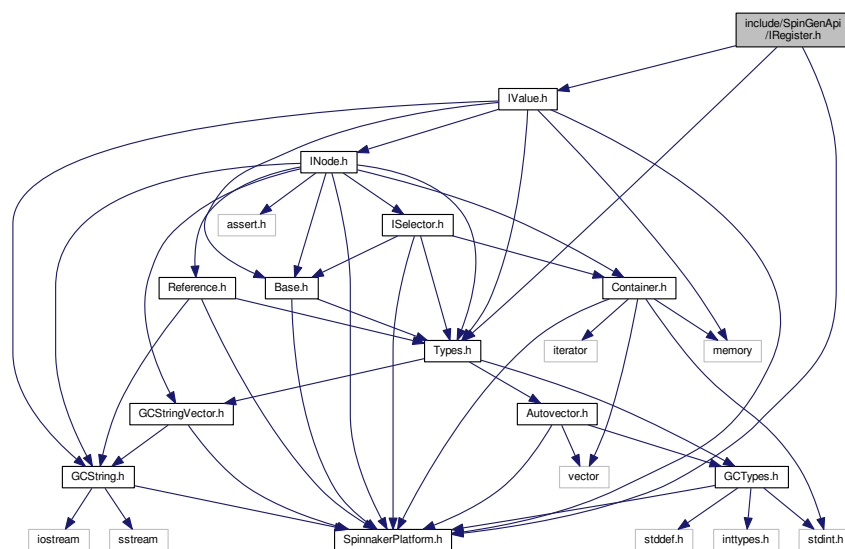
*Interface for replaying write commands on a port.*

- [interface SPINNAKER\\_API\\_ABSTRACT bool Invalidate = true\) = 0](#)
- [interface SPINNAKER\\_API\\_ABSTRACT IPortRecorder](#)

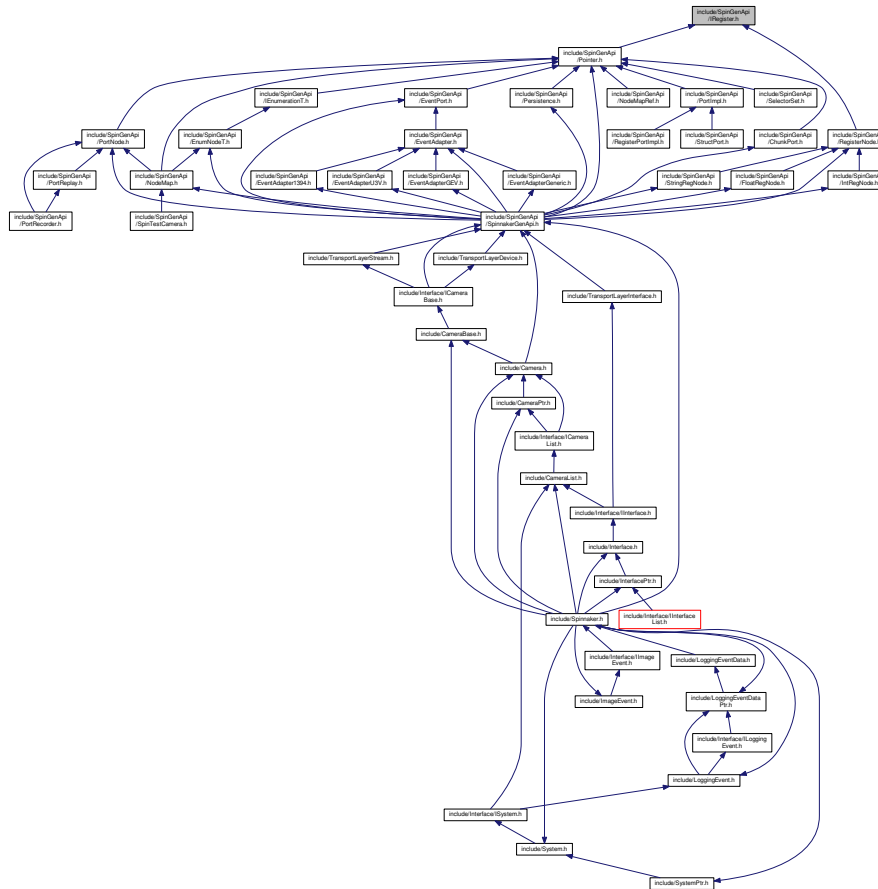
*Interface for recording write commands on a port.*

## 11.93 include/SpinGenApi/IRegister.h File Reference

Include dependency graph for IRegister.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

- virtual void [Get](#) (uint8\_t \*pBuffer, int64\_t Length, bool Verify=false, bool IgnoreCache=false)=0  
*Fills a buffer with the register's contents.*
- virtual int64\_t [GetLength](#) ()=0  
*Retrieves the Length of the register [Bytes].*
- virtual int64\_t [GetAddress](#) ()=0  
*Retrieves the Address of the register.*

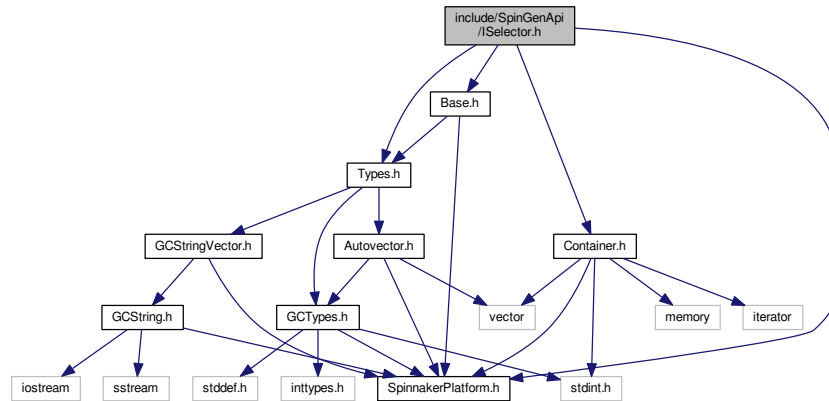
## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IRegister](#)  
*Interface for registers.*

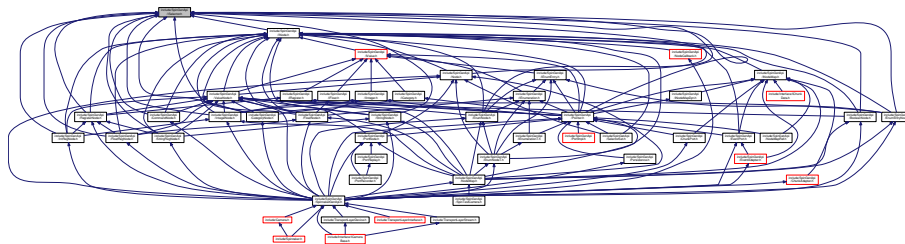


## 11.94 include/SpinGenApi/ISelector.h File Reference

Include dependency graph for ISelector.h:



This graph shows which files directly or indirectly include this file:



### Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

### Functions

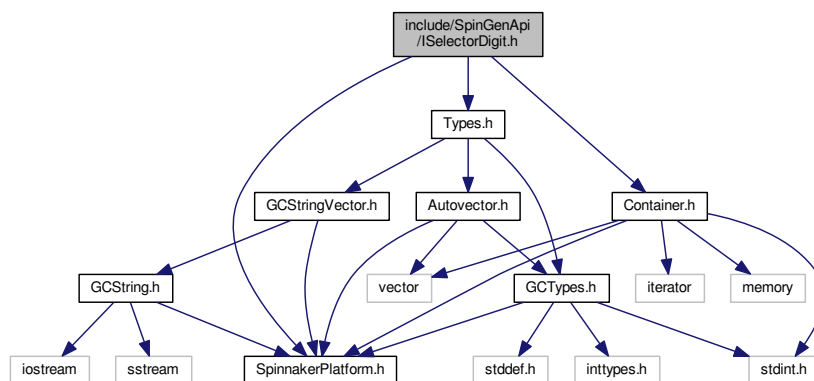
- virtual void [GetSelectedFeatures](#) (FeatureList\_t &) const =0  
*retrieve the group of selected features*
- virtual void [GetSelectingFeatures](#) (FeatureList\_t &) const =0  
*retrieve the group of features selecting this node*

### Variables

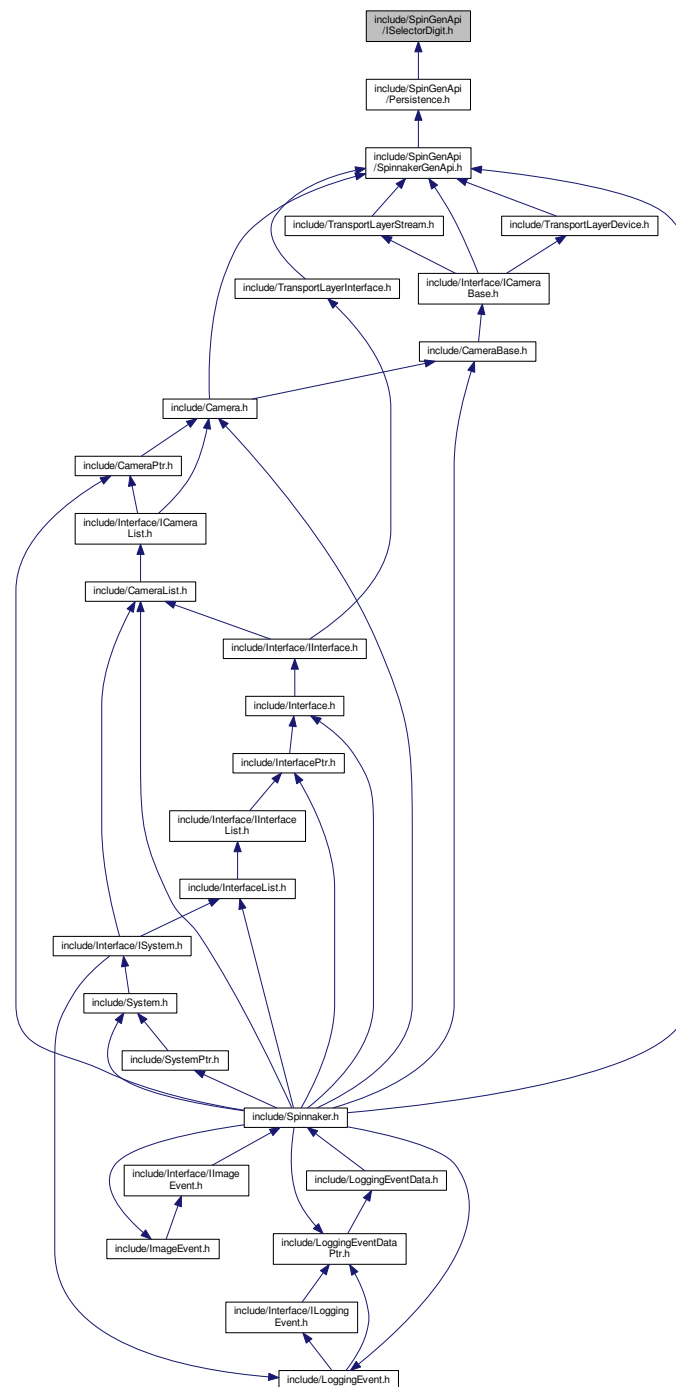
- [interface SPINNAKER\\_API\\_ABSTRACT ISelector](#)  
*Interface for groups of features selected by a single one.*

## 11.95 include/SpinGenApi/ISelectorDigit.h File Reference

Include dependency graph for ISelectorDigit.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

- virtual bool [SetNext](#) (bool Tick=true)=0

*Sets digit to next value.*

- virtual void [Restore](#) ()=0

*Restores the selectors' values found at creation.*

- virtual GenICam::gcstring [ToString](#) ()=0

*Returns a string representation of the digit.*

- virtual void [GetSelectorList](#) (FeatureList\_t &SelectorList, bool Incremental=false)=0

*Retrieves an ordered list of selectors.*

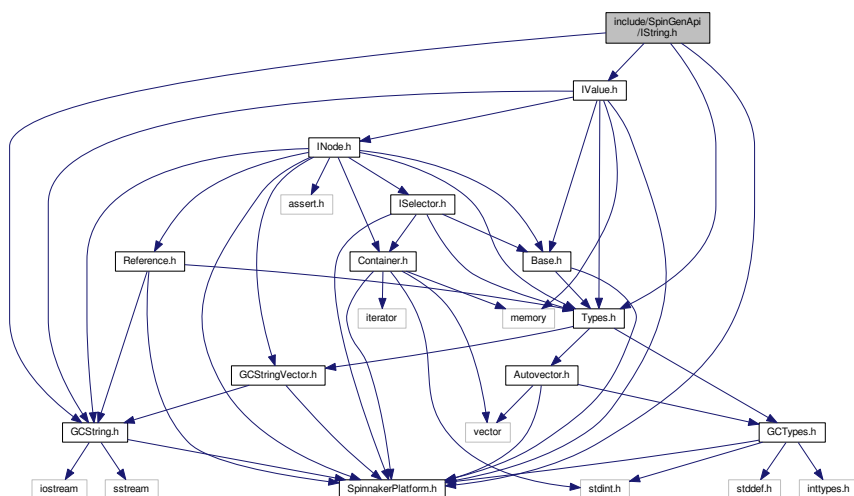
## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT ISelectorDigit](#)

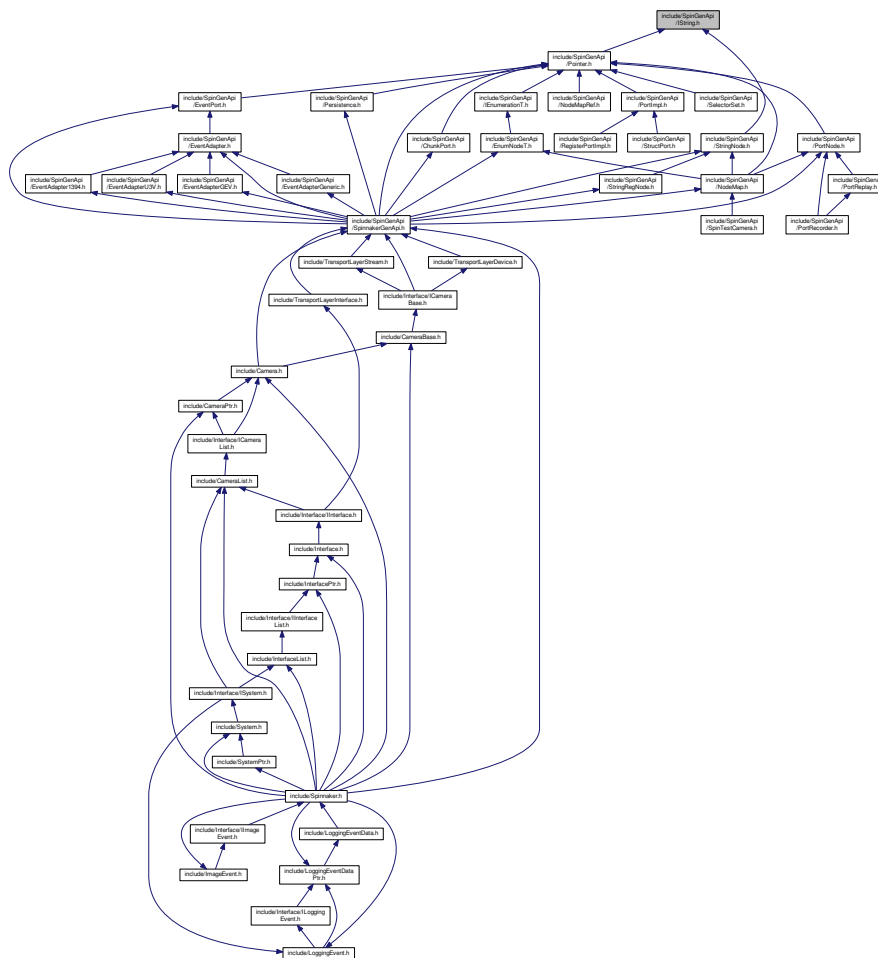
*Interface of a "digit" of the "counter" formed by the selector set.*

## 11.96 include/SpinGenApi/IString.h File Reference

Include dependency graph for IString.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- Spinnaker
- Spinnaker::GenApi

## Functions

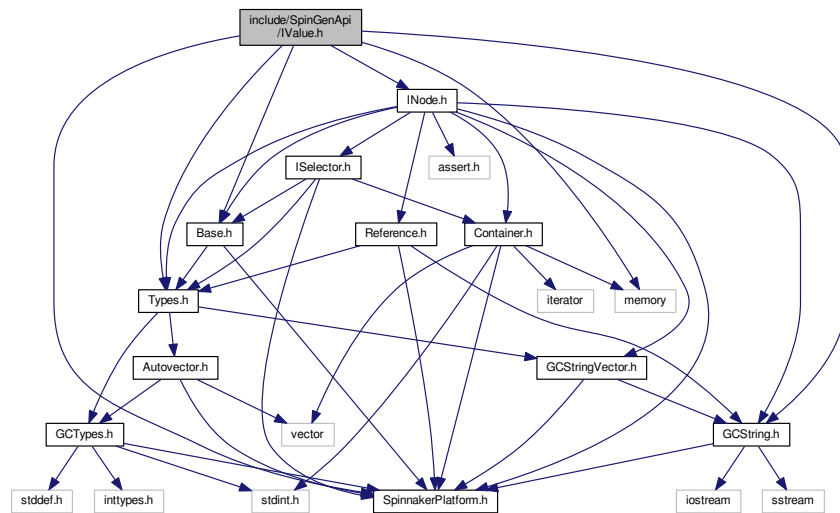
- virtual IEnumeration & **operator=** (const GenICam::gcstring &ValueStr)=0  
*Set string node value.*
- virtual bool **GetValue** (bool Verify=false, bool IgnoreCache=false) const =0  
*Get node value.*
- virtual bool **operator()** () const  
*Get node value.*
- virtual GenICam::gcstring **operator\*** ()=0  
*Get string node value.*
- virtual int64\_t **GetMaxLength** ()=0  
*Retrieves the maximum length of the string in bytes.*

## Variables

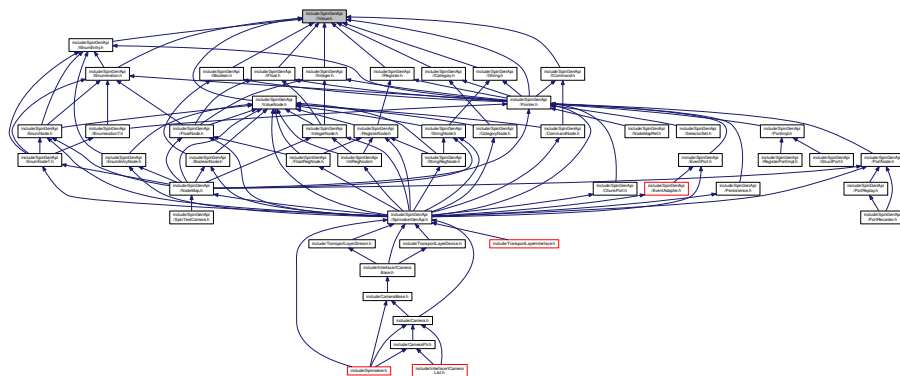
- [interface SPINNAKER\\_API\\_ABSTRACT IString](#)  
*Interface for string properties.*

## 11.97 include/SpinGenApi/IValue.h File Reference

Include dependency graph for IValue.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

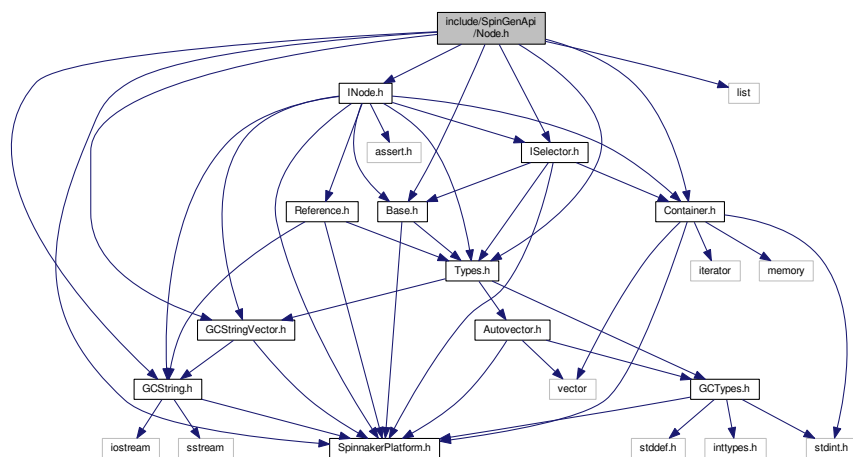
- virtual GenICam::gcstring [ToString](#) (bool Verify=false, bool IgnoreCache=false)=0  
*Get content of the node as string.*
- virtual void [FromString](#) (const GenICam::gcstring &ValueStr, bool Verify=true)=0  
*Set content of the node as string.*
- virtual bool [IsValueCacheValid](#) () const =0  
*Checks if the value comes from cache or is requested from another node.*

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IValue](#)  
*Interface for value properties.*

## 11.98 include/SpinGenApi/Node.h File Reference

Include dependency graph for Node.h:

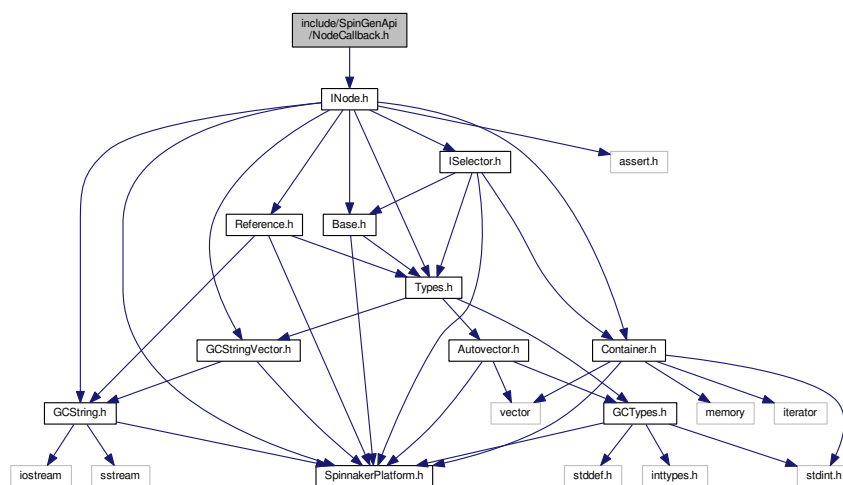






## 11.99 include/SpinGenApi/NodeCallback.h File Reference

Include dependency graph for NodeCallback.h:





## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Enumerations

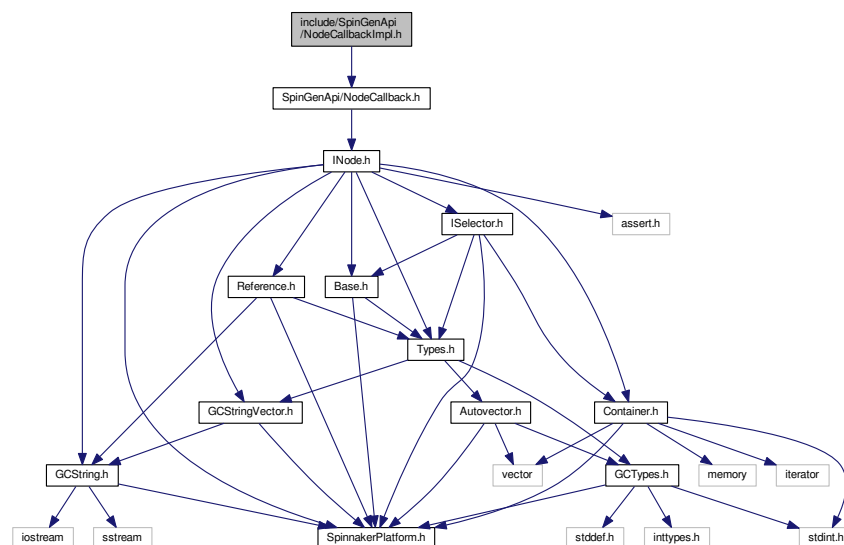
- enum [ECallbackType](#) {  
[cbPostInsideLock](#) = 1,  
[cbPostOutsideLock](#) = 2 }  
*the type of callback*

## Functions

- template<class Function >  
 CNodeCallback \* [make\\_NodeCallback](#) (INode \*pNode, Function function, ECallbackType CallbackType)  
*make a new callback object for C functions*
- template<class Function >  
 intptr\_t [Register](#) (INode \*pNode, Function f, ECallbackType CallbackType=cbPostInsideLock)  
*Register a C-function as a callback.*
- template<class Client , class Member >  
 CNodeCallback \* [make\\_NodeCallback](#) (INode \*pNode, Client &client, Member member, ECallbackType CallbackType)  
*make a new callback object for member functions*
- template<class Client , class Member >  
 intptr\_t [Register](#) (INode \*pNode, Client &c, Member m, ECallbackType CallbackType=cbPostInsideLock)  
*Register a C++-member function a callback.*
- [SPINNAKER\\_API](#) void [Deregister](#) (GenApi::CallbackHandleType pCallbackInfo)  
*Unregistering callback by handle.*

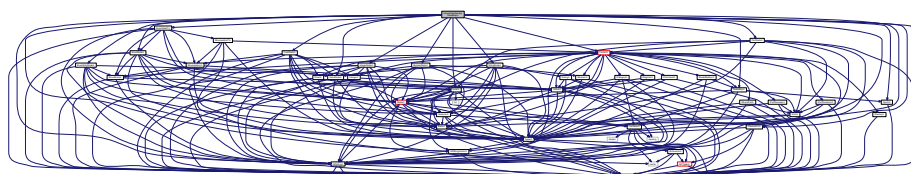
## 11.100 include/SpinGenApi/NodeCallbackImpl.h File Reference

Include dependency graph for NodeCallbackImpl.h:

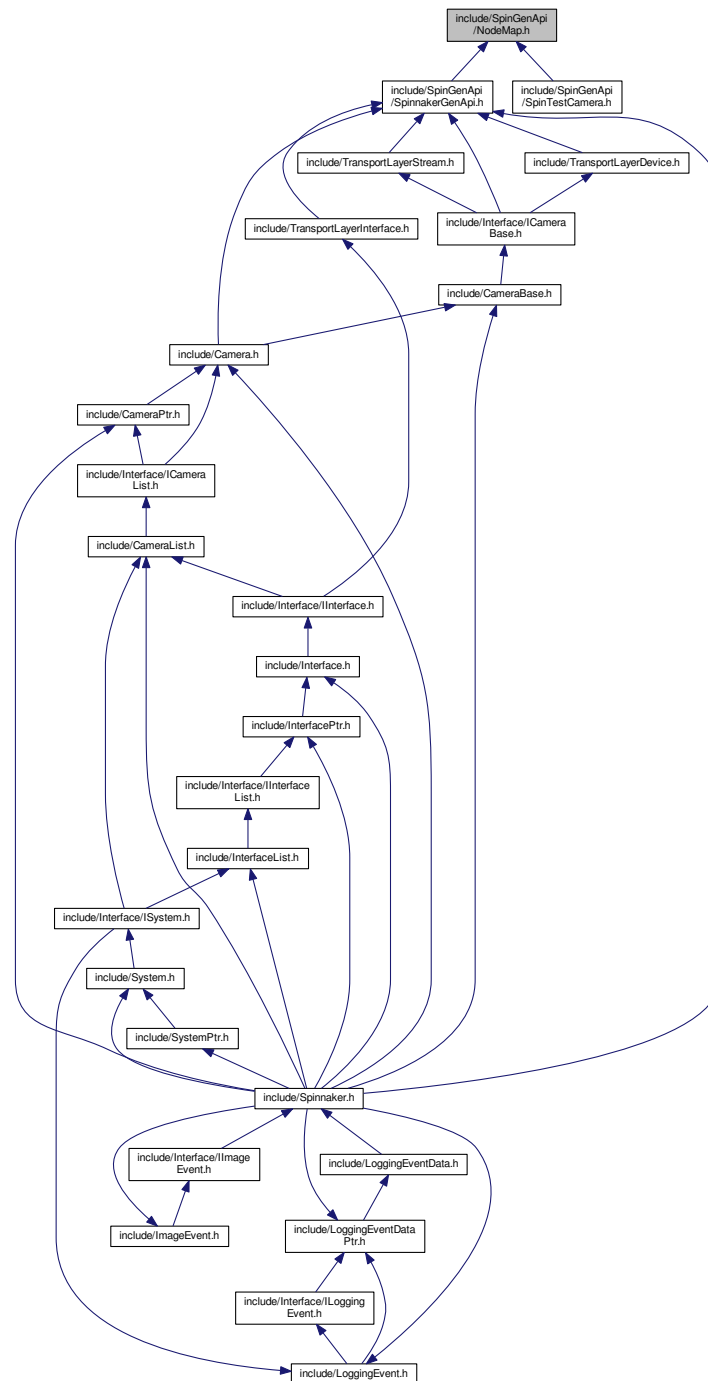


## 11.101 include/SpinGenApi/NodeMap.h File Reference

Include dependency graph for NodeMap.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [NodeMap](#)

*Smart pointer template for NodeMaps with create function.*

## Namespaces

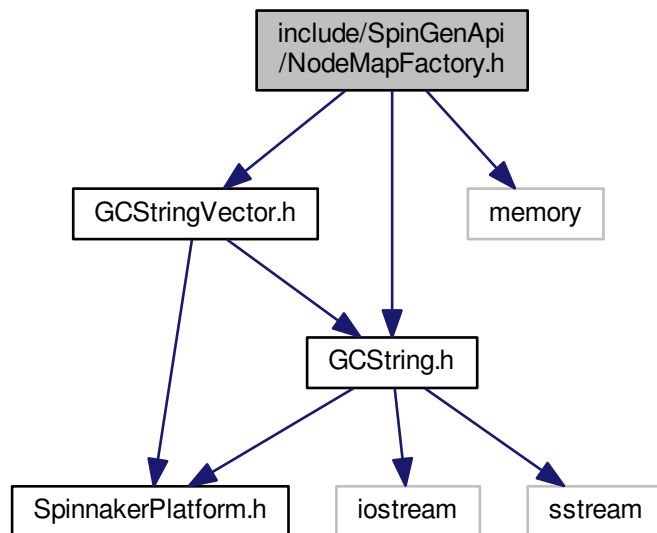
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

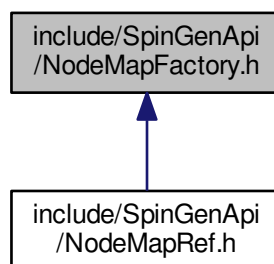
- typedef NodeMap [CNodeMapRef](#)

## 11.102 include/SpinGenApi/NodeMapFactory.h File Reference

Include dependency graph for NodeMapFactory.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [CNodeMapFactory](#)  
The node map factory is used for creating node maps from camera description files.
- struct [CNodeMapFactory::NodeStatistics\\_t](#)

## Namespaces

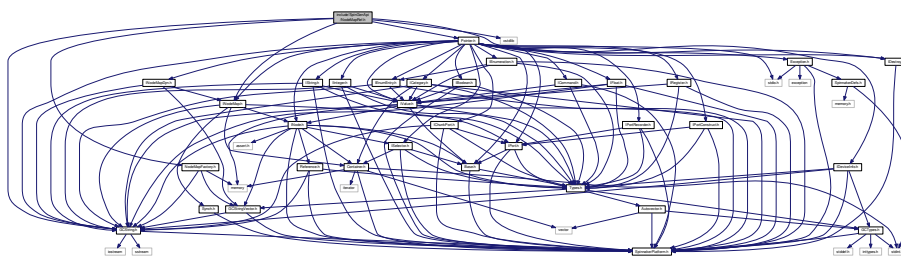
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Enumerations

- enum [ECacheUsage\\_t](#) {  
  [CacheUsage\\_Automatic](#),  
  [CacheUsage\\_ForceWrite](#),  
  [CacheUsage\\_ForceRead](#),  
  [CacheUsage\\_Ignore](#) }  
Lists the cache usage strategies.
- enum [EContentType\\_t](#) {  
  [ContentType\\_Xml](#),  
  [ContentType\\_ZippedXml](#) }  
Lists the processable file types.

## 11.103 include/SpinGenApi/NodeMapRef.h File Reference

Include dependency graph for NodeMapRef.h:



## Classes

- class [CNodeMapRefT< TCameraParams >](#)  
Smartpointer template for NodeMaps with create function.
- class [CGeneric\\_XMLLoaderParams](#)  
Empty base class used by class [CNodeMapRef](#) as generic template argument.
- class [CNodeMapRef](#)  
Smartpointer for NodeMaps with create function.

## Namespaces

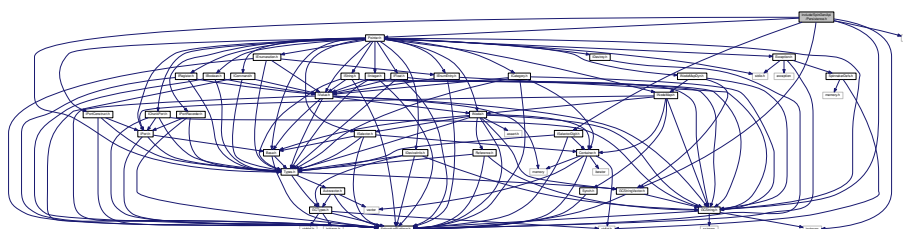
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

- [SPINNAKER\\_API](#) [IDestroy](#) \* [CastToIDestroy](#) (INodeMap \*pNodeMap)  
*makes sure the dynamic\_cast operator is implemented in the DLL (due to a Linux bug)*
- [template](#)<class TCameraParams >  
[void](#) [\\_LoadXMLFromFile](#) (const GenICam::gcstring &FileName)
- [template](#)<class TCameraParams >  
[void](#) [\\_LoadXMLFromZIPFile](#) (const GenICam::gcstring &ZipFileName)
- [template](#)<class TCameraParams >  
[void](#) [\\_LoadXMLFromFileInject](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)
- [template](#)<class TCameraParams >  
[void](#) [\\_LoadXMLFromString](#) (const GenICam::gcstring &XMLData)
- [template](#)<class TCameraParams >  
[void](#) [\\_LoadXMLFromZIPData](#) (const void \*zipData, size\_t zipSize)
- [template](#)<class TCameraParams >  
[void](#) [\\_LoadXMLFromStringInject](#) (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)
- [template](#)<class TCameraParams >  
[void](#) [\\_GetSupportedSchemaVersions](#) (GenICam::gcstring\_vector &SchemaVersions)
- [template](#)<class TCameraParams >  
[GenICam::gcstring](#) [\\_GetDeviceName](#) ()
- [template](#)<class TCameraParams >  
[void](#) [\\_Poll](#) (int64\_t ElapsedTime)
- [template](#)<class TCameraParams >  
[void](#) [\\_GetNodes](#) (NodeList\_t &Nodes)
- [template](#)<class TCameraParams >  
[INode](#) \* [\\_GetNode](#) (const GenICam::gcstring &key)
- [template](#)<class TCameraParams >  
[void](#) [\\_InvalidateNodes](#) ()
- [template](#)<class TCameraParams >  
[bool](#) [\\_Connect](#) (IPort \*pPort, const GenICam::gcstring &PortName)
- [template](#)<class TCameraParams >  
[bool](#) [\\_Connect](#) (IPort \*pPort)
- [template](#)<class TCameraParams >  
[bool](#) [\\_ClearXMLCache](#) ()

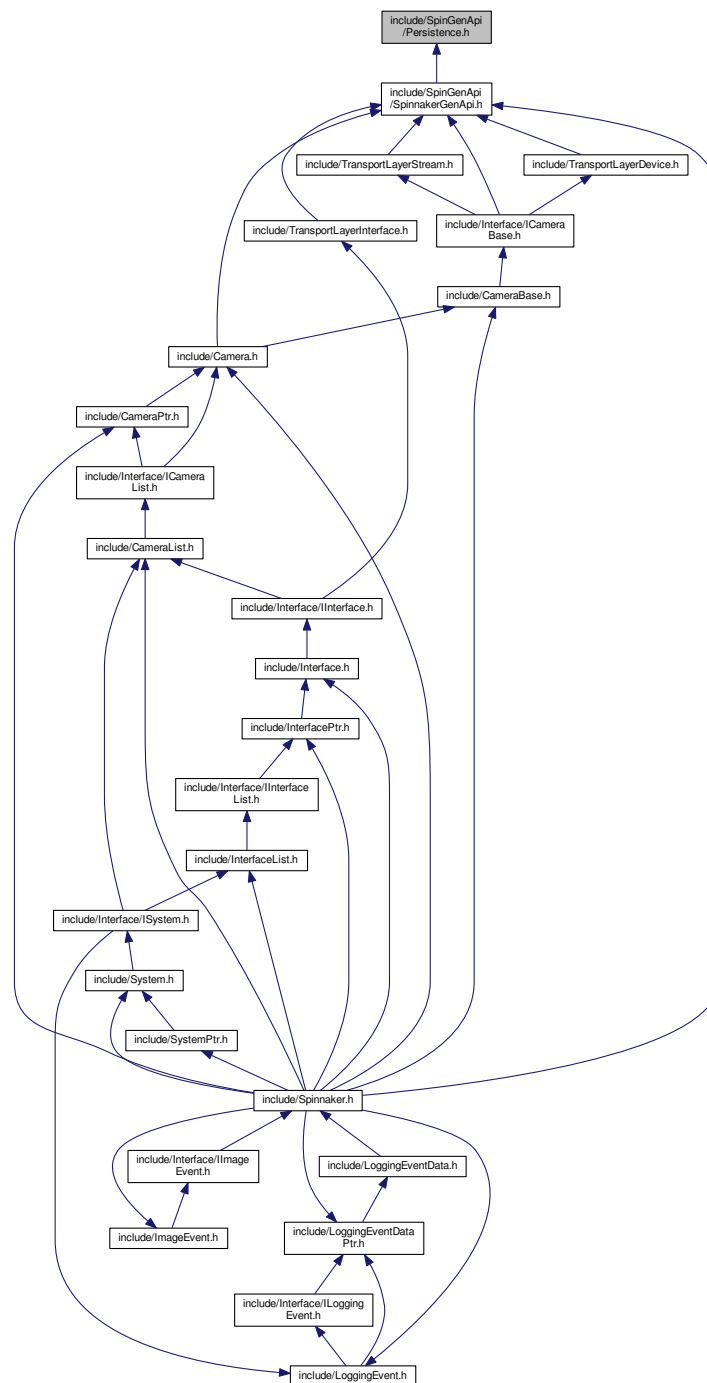
## 11.104 include/SpinGenApi/Persistence.h File Reference

Include dependency graph for Persistence.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [CFeatureBag](#)

*Bag holding streamable features of a nodetree.*

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

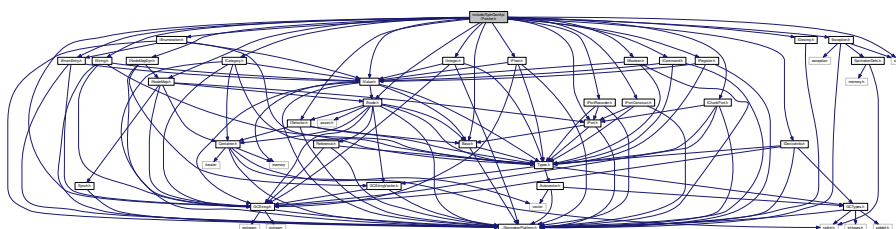
- virtual void [PersistFeature](#) (IValue &item)=0  
*Stores a feature.*
- [SPINNAKER\\_API](#) std::istream & [EatComments](#) (std::istream &is)  
*Helper function ignoring lines starting with comment character '#'.  
Reads in persistent data from a stream.*
- [SPINNAKER\\_API](#) std::istream & [operator>>](#) (std::istream &is, CFeatureBag &FeatureBag)  
*Reads in persistent data from a stream.*
- [SPINNAKER\\_API](#) std::ostream & [operator<<](#) (std::ostream &os, const CFeatureBag &FeatureBag)  
*writes out persistent data to a stream*

## Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IPersistScript](#)  
*Basic interface to persist values to.*

## 11.105 include/SpinGenApi/Pointer.h File Reference

Include dependency graph for Pointer.h:



[illegible]

- class `CPointer< T, B >`  
*Encapsulates a `GenApi` pointer dealing with the `dynamic_cast` automatically.*
- class `CFloatPtr`  
*SmartPointer for `IFloat` interface pointer.*

- Spinnaker
- Spinnaker::GenApi

- typedef CPointer< IBase > CBasePtr  
*SmartPointer for IBase interface pointer.*
- typedef CPointer< INode, IBase > CNodePtr  
*SmartPointer for INode interface pointer.*

- typedef CPointer< IValue > CValuePtr  
*SmartPointer for IValue interface pointer.*
- typedef CPointer< ICategory > CCategoryPtr  
*SmartPointer for ICategory interface pointer.*
- typedef CPointer< IBoolean > CBooleanPtr  
*SmartPointer for IBoolean interface pointer.*
- typedef CPointer< IInteger > CIntegerPtr  
*SmartPointer for IInteger interface pointer.*
- typedef CPointer< IString > CStringPtr  
*SmartPointer for IString interface pointer.*
- typedef CPointer< IRegister > CRegisterPtr  
*SmartPointer for IRegister interface pointer.*
- typedef CPointer< IEnumeration > CEnumerationPtr  
*SmartPointer for IEnumeration interface pointer.*
- typedef CPointer< IEnumEntry > CEnumEntryPtr  
*SmartPointer for IEnumEntry interface pointer.*
- typedef CPointer< IPort > CPortPtr  
*SmartPointer for IPort interface pointer.*
- typedef CPointer< IPortReplay > CPortReplayPtr  
*SmartPointer for IPortReplay interface pointer.*
- typedef CPointer< IPortRecorder > CPortRecorderPtr  
*SmartPointer for IPortRecorder interface pointer.*
- typedef CPointer< IPortWriteList, IPortWriteList > CPortWriteListPtr  
*SmartPointer for IPortWriteList interface pointer.*
- typedef CPointer< IChunkPort > CChunkPortPtr  
*SmartPointer for IChunkPort interface pointer.*
- typedef CPointer< INodeMap, INodeMap > CNodeMapPtr  
*SmartPointer for INodeMap interface pointer.*
- typedef CPointer< INodeMapDyn, INodeMap > CNodeMapDynPtr  
*SmartPointer for INodeMapDyn interface pointer.*
- typedef CPointer< IDeviceInfo, INodeMap > CDeviceInfoPtr  
*SmartPointer for IDeviceInfo interface pointer.*
- typedef CPointer< ISelector > CSelectorPtr  
*SmartPointer for ISelector interface pointer.*
- typedef CPointer< ICommand > CCommandPtr  
*SmartPointer for ICommand interface pointer.*
- typedef CPointer< IPortConstruct > CPortConstructPtr  
*SmartPointer for IPortConstruct interface pointer.*

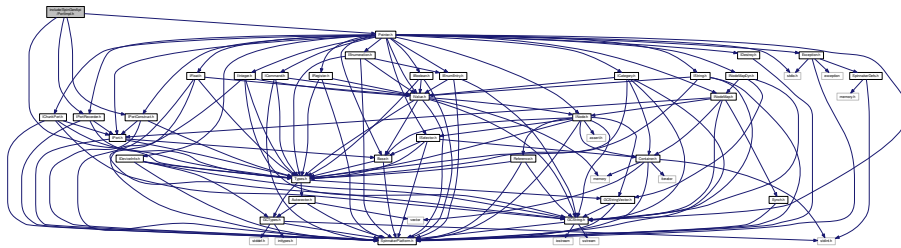
## Functions

- template<class T , class B >  
bool IsReadable (const Spinnaker::GenApi::CPointer< T, B > &ptr)  
*Checks if a node is readable.*
- template<class T , class B >  
bool IsWritable (const Spinnaker::GenApi::CPointer< T, B > &ptr)  
*Checks if a node is Writable.*
- template<class T , class B >  
bool IsImplemented (const Spinnaker::GenApi::CPointer< T, B > &ptr)  
*Checks if a node is Implemented.*

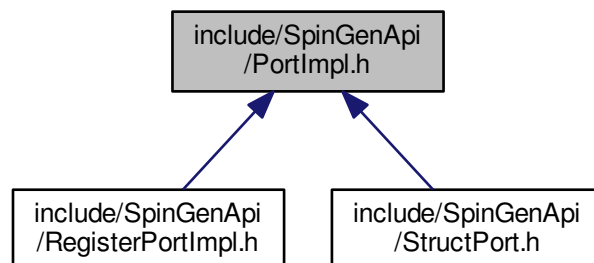
- `template<class T, class B>`  
`bool IsAvailable (const Spinnaker::GenApi::CPointer< T, B> &ptr)`  
*Checks if a node is Available.*
- `GenICam::gcstring GetInterfaceName (IBase *pBase)`  
*Returns the name of the main interface as string DEPRICATED, use `IBase::GetPrincipalInterfaceType()` instead.*

## 11.106 include/SpinGenApi/PortImpl.h File Reference

Include dependency graph for PortImpl.h:



This graph shows which files directly or indirectly include this file:



### Classes

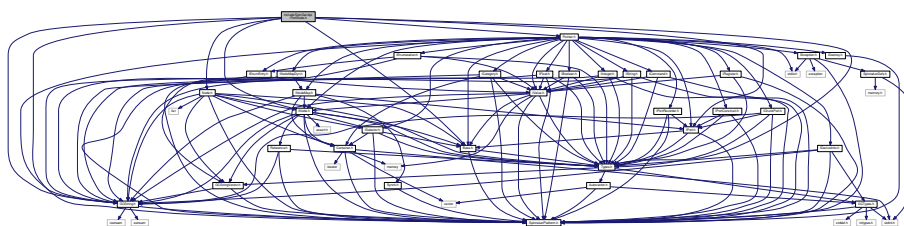
- class `CPortImpl`  
*Standard implementation for a port.*

### Namespaces

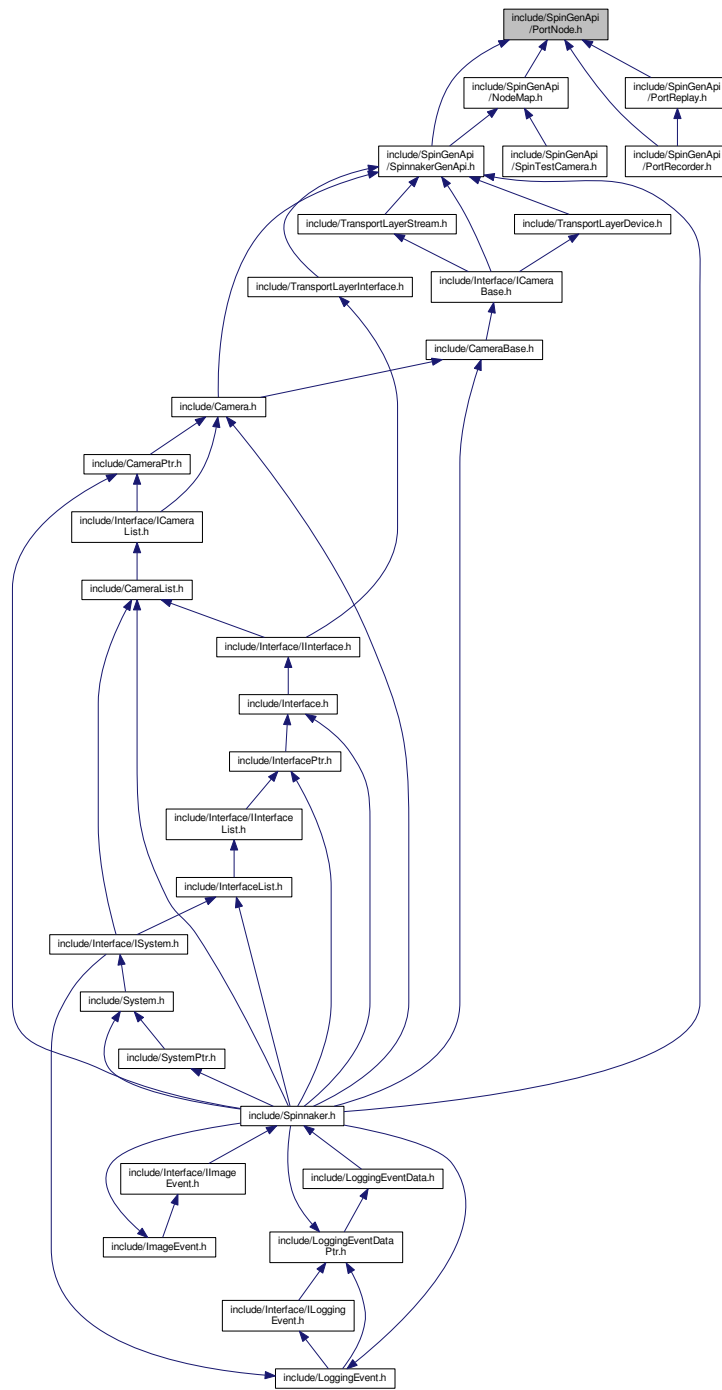
- `Spinnaker`
- `Spinnaker::GenApi`

## 11.107 include/SpinGenApi/PortNode.h File Reference

Include dependency graph for PortNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [PortNode](#)  
*Interface* for value properties.

## Namespaces

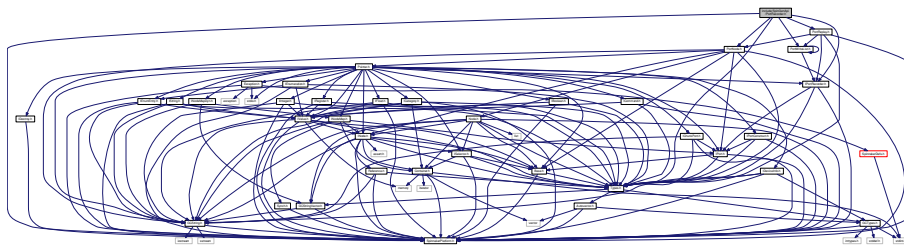
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- typedef PortNode [CPortRef](#)

## 11.108 include/SpinGenApi/PortRecorder.h File Reference

Include dependency graph for PortRecorder.h:



## Classes

- class [PortRecorder](#)  
*Interface for recording write commands on a port.*

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

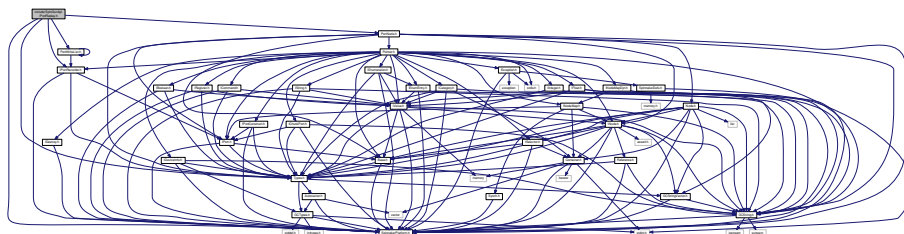
## Typedefs

- typedef PortRecorder [CPortRecorderRef](#)  
*Reference to an IPortRecorder pointer.*

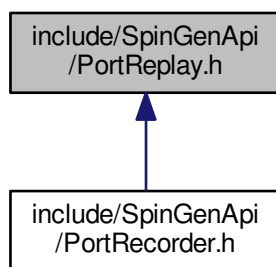


## 11.109 include/SpinGenApi/PortReplay.h File Reference

Include dependency graph for PortReplay.h:



This graph shows which files directly or indirectly include this file:



### Classes

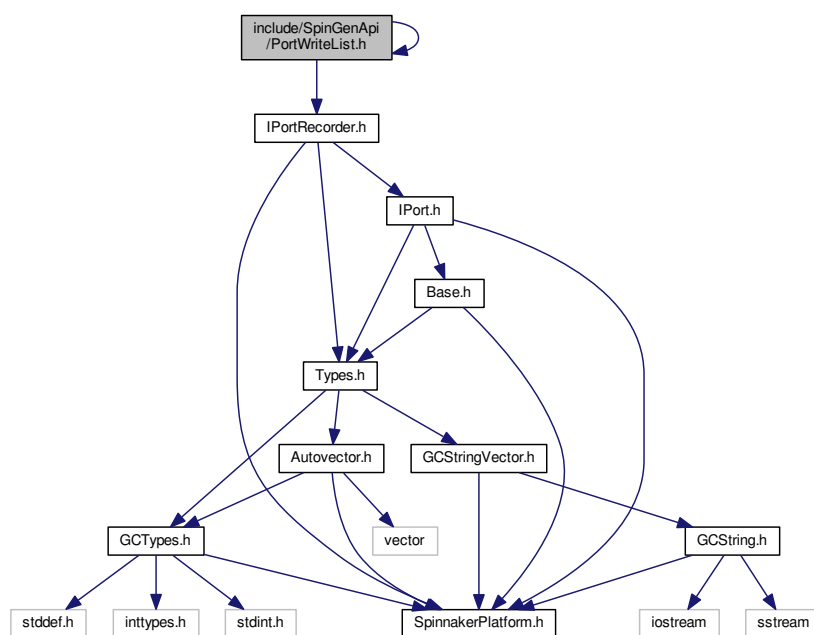
- class [PortReplay](#)  
*Interface for replaying write commands on a port.*

### Namespaces

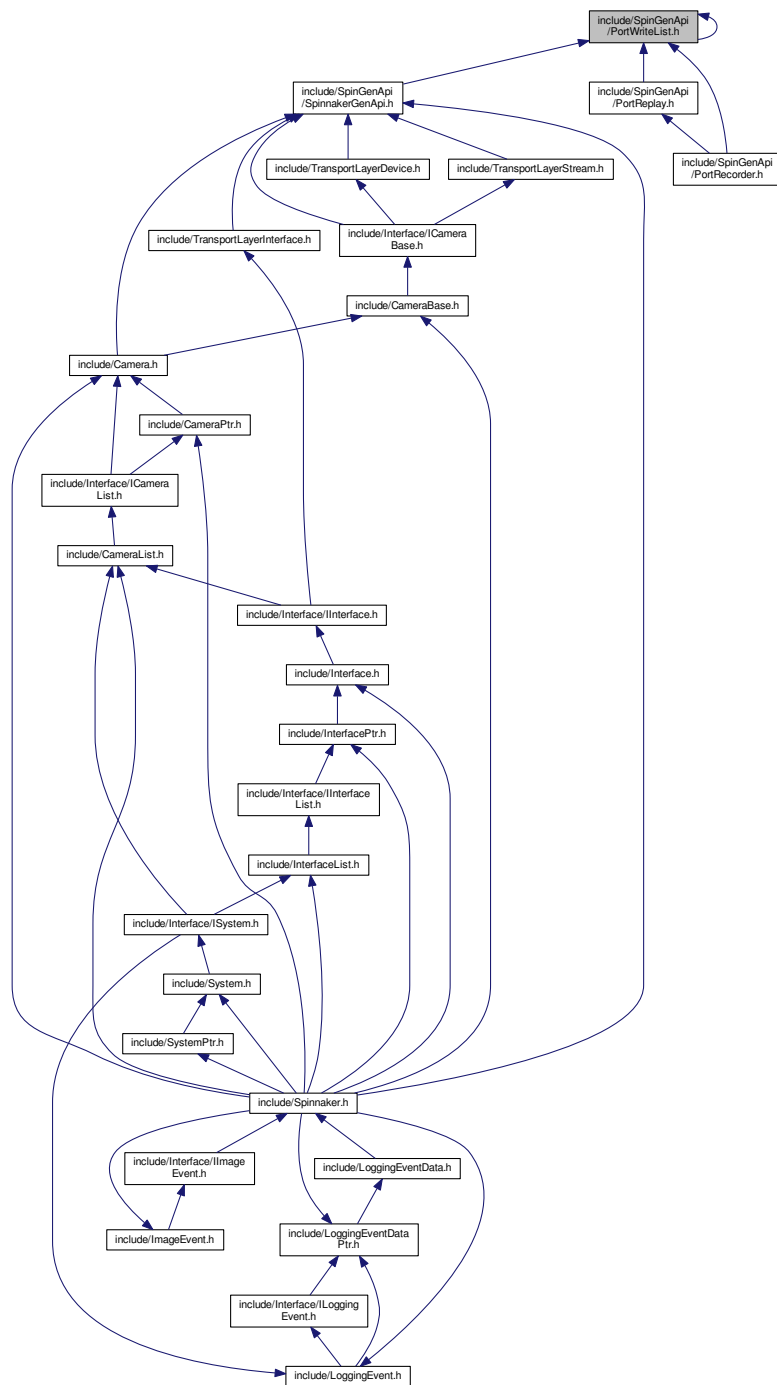
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.110 include/SpinGenApi/PortWriteList.h File Reference

Include dependency graph for PortWriteList.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [CPortWriteList](#)

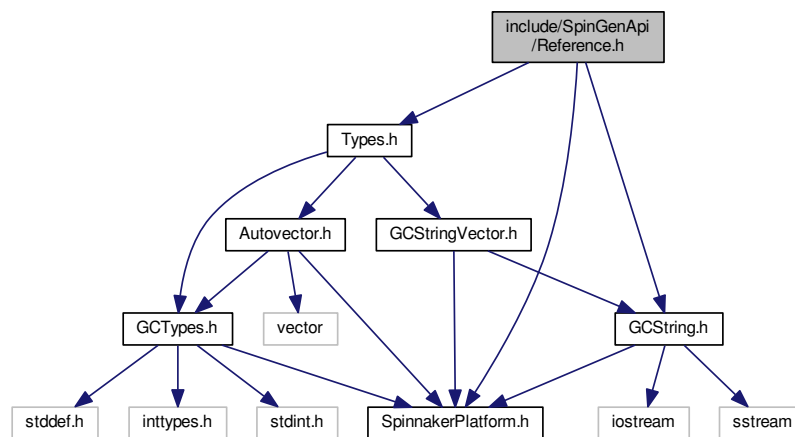
*Container holding a list of port write commands.*

## Namespaces

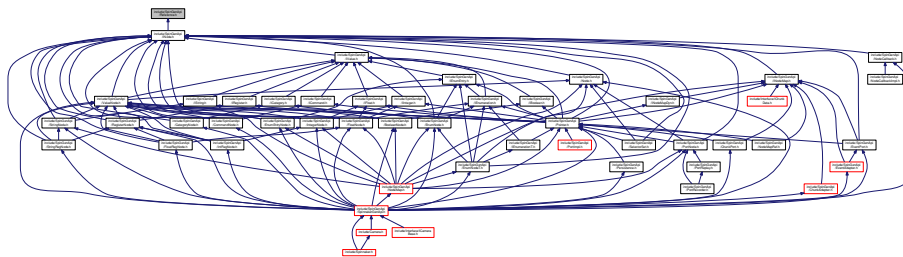
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

### 11.111 include/SpinGenApi/Reference.h File Reference

Include dependency graph for Reference.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

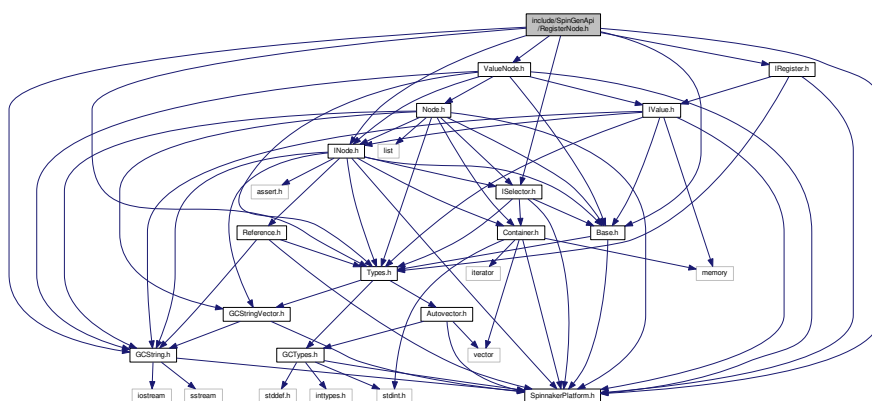
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

- virtual void [SetNumEnums](#) (int NumEnums)=0  
*sets the number of enum values*

## 11.112 include/SpinGenApi/RegisterNode.h File Reference

Include dependency graph for RegisterNode.h:





## Namespaces

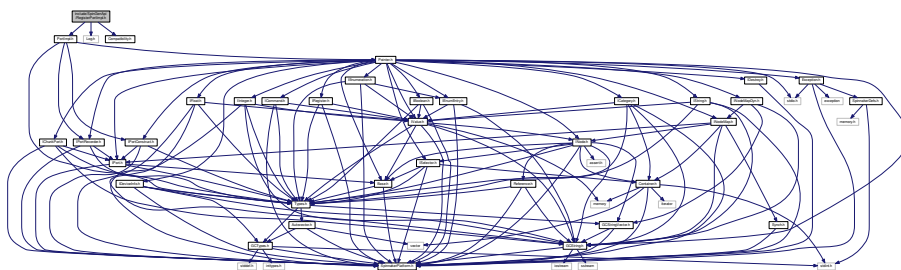
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- typedef RegisterNode [CRegisterRef](#)

## 11.113 include/SpinGenApi/RegisterPortImpl.h File Reference

Include dependency graph for RegisterPortImpl.h:



## Classes

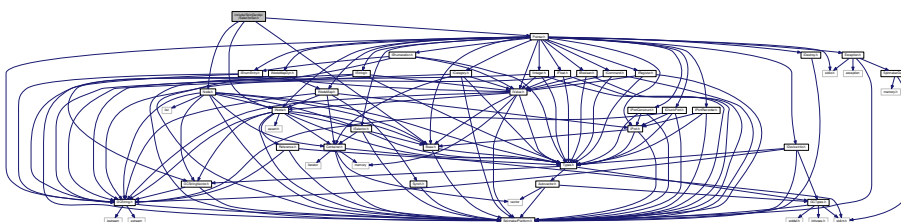
- class [CRegisterPortImpl](#)  
Standard implementation for a port using a register based transport layer.

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.114 include/SpinGenApi/SelectorSet.h File Reference

Include dependency graph for SelectorSet.h:



## Classes

- class [CSelectorSet](#)

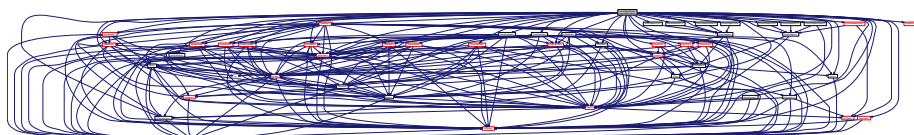
*The set of selectors selecting a given node.*

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

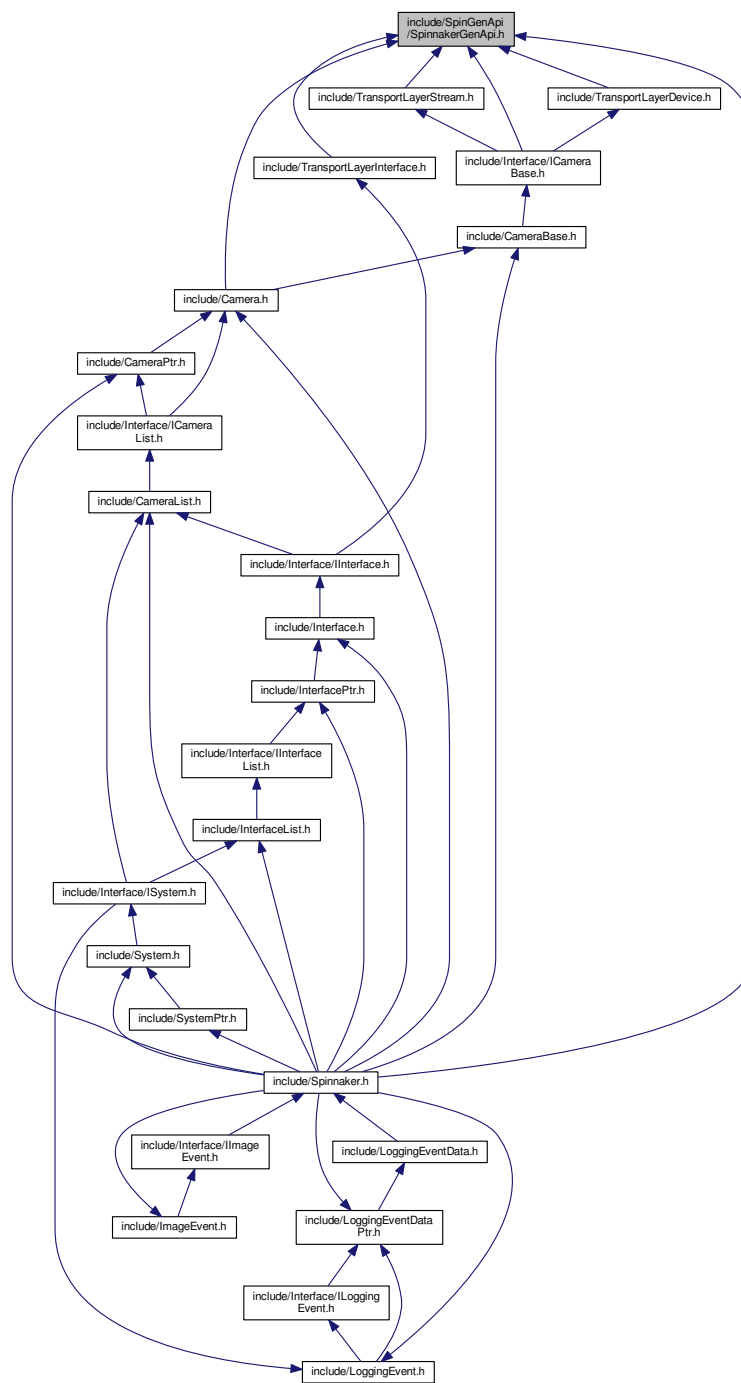
### 11.115 include/SpinGenApi/SpinnakerGenApi.h File Reference

Include dependency graph for SpinnakerGenApi.h:



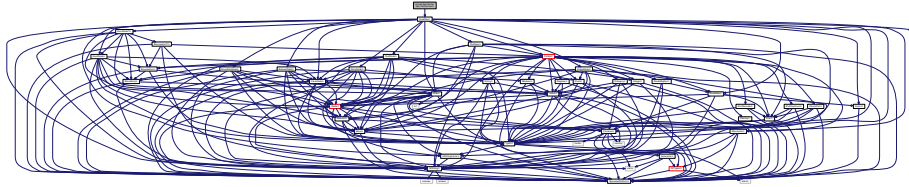


This graph shows which files directly or indirectly include this file:



## 11.116 include/SpinGenApi/SpinTestCamera.h File Reference

Include dependency graph for SpinTestCamera.h:



### Classes

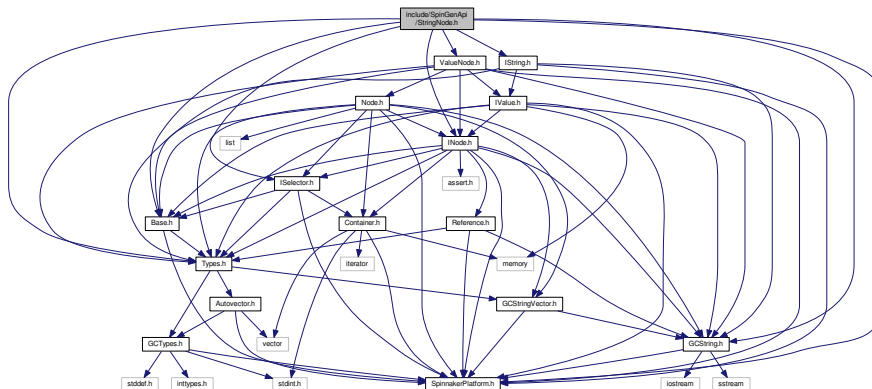
- class [SpinTestCamera](#)

### Namespaces

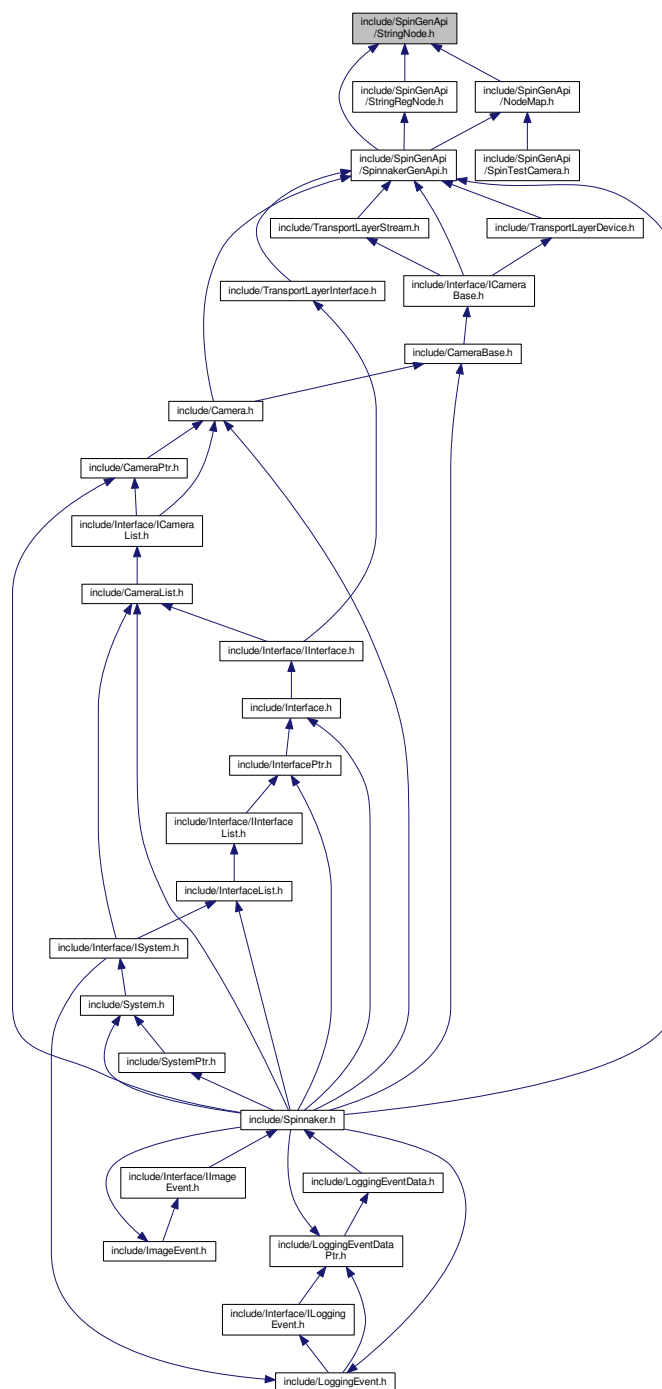
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.117 include/SpinGenApi/StringNode.h File Reference

Include dependency graph for StringNode.h:

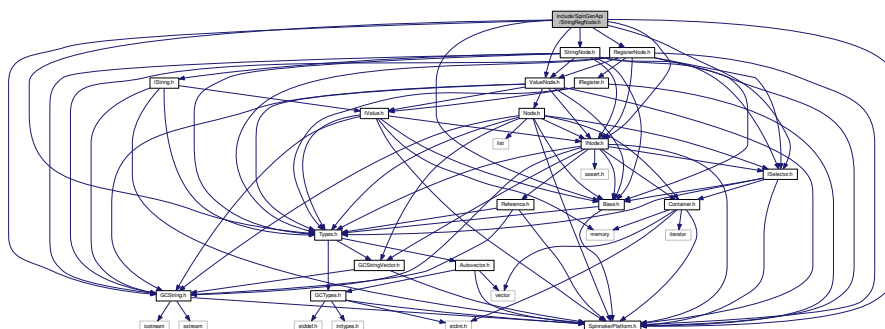


This graph shows which files directly or indirectly include this file:

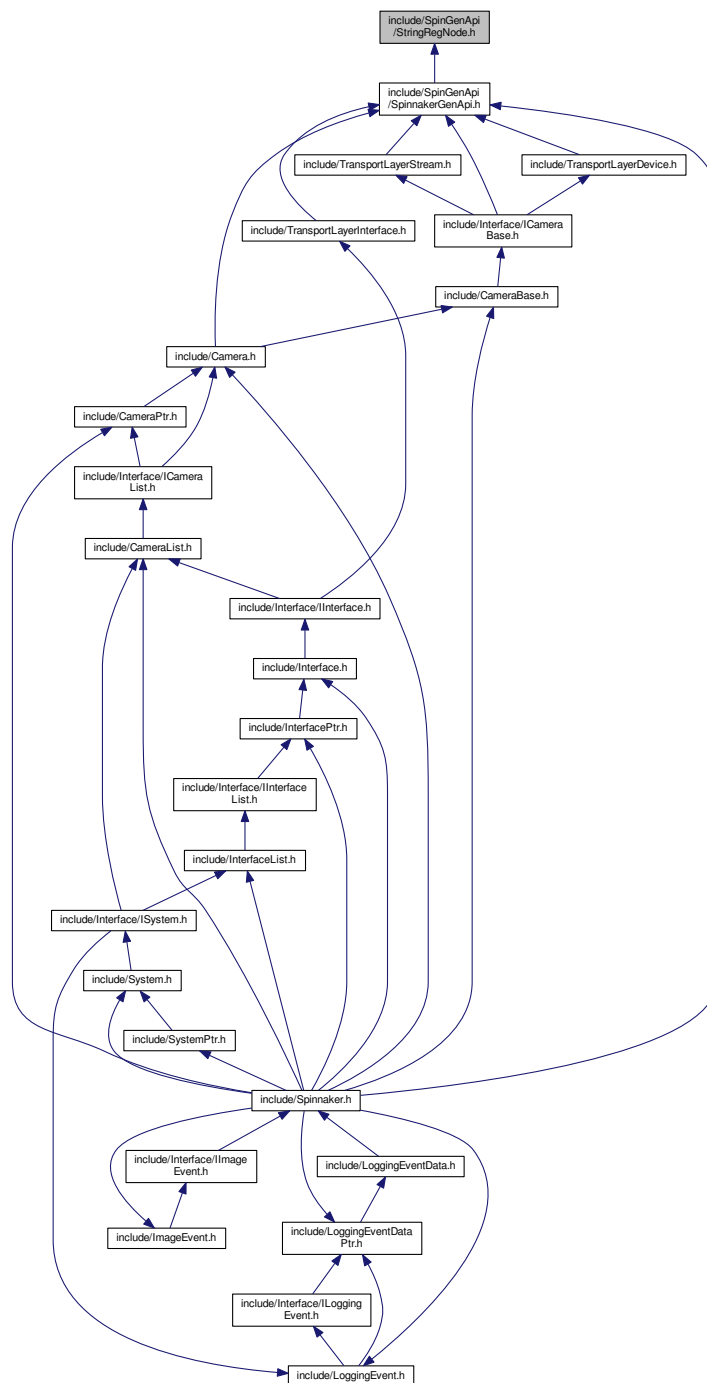


## Classes

- class [StringNode](#)  
*Interface for string properties.*



This graph shows which files directly or indirectly include this file:



## Classes

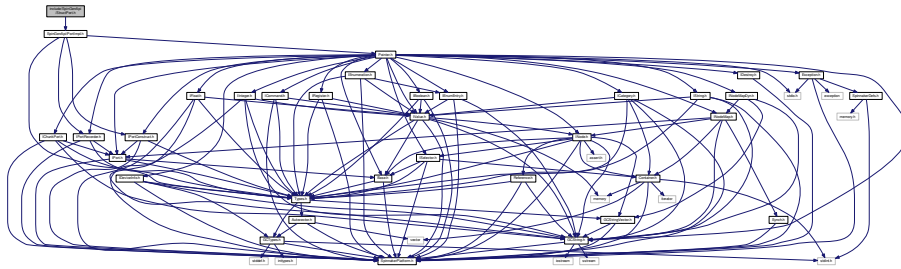
- class **StringRegNode**  
*Interface for string properties.*

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

### 11.119 include/SpinGenApi/StructPort.h File Reference

Include dependency graph for StructPort.h:



## Classes

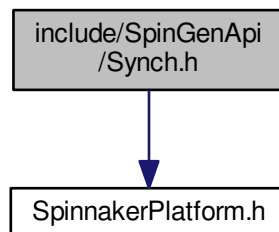
- class [CTestPortStruct< CDataStruct >](#)  
*Implements a register spaces based on a C++ struct.*

## Namespaces

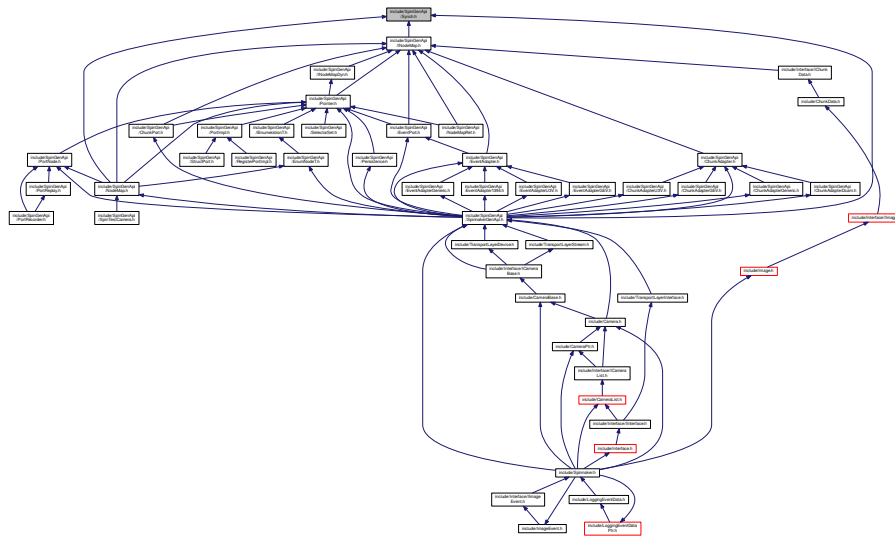
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

### 11.120 include/SpinGenApi/Synch.h File Reference

Include dependency graph for Synch.h:



This graph shows which files directly or indirectly include this file:



## Classes

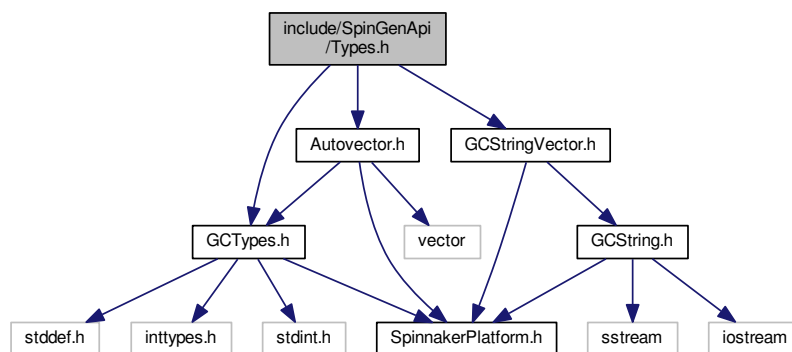
- class [CLock](#)  
A lock class.
- class [CLockEx](#)  
This class is for testing purposes only.
- class [AutoLock](#)

## Namespaces

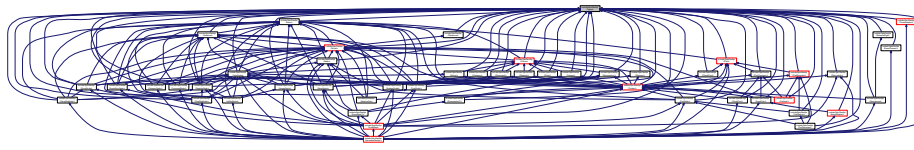
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 11.121 include/SpinGenApi/Types.h File Reference

Include dependency graph for Types.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Macros

- `#define` [interface](#) struct
- `#define` [\\_UndefinedRepresentation](#) \_UndefinedRepresentation

## Typedefs

- `typedef` GenICam::gcstring\_vector [StringList\\_t](#)  
*A list of strings.*

## Enumerations

- `enum` [ESign](#) {  
[Signed](#),  
[Unsigned](#),  
[\\_UndefinedSign](#) }  
*signed or unsigned integers*
- `enum` [EAccessMode](#) {  
[NI](#),  
[NA](#),  
[WO](#),  
[RO](#),  
[RW](#),  
[\\_UndefinedAccesMode](#),  
[\\_CycleDetectAccesMode](#) }  
*access mode of a node*
- `enum` [EVisibility](#) {  
[Beginner](#) = 0,  
[Expert](#) = 1,  
[Guru](#) = 2,  
[Invisible](#) = 3,  
[\\_UndefinedVisibility](#) = 99 }  
*recommended visibility of a node*
- `enum` [ECachingMode](#) {  
[NoCache](#),  
[WriteThrough](#),  
[WriteAround](#),  
[\\_UndefinedCachingMode](#) }



- caching mode of a register*
- enum [ERepresentation](#) {  
[Linear](#),  
[Logarithmic](#),  
[Boolean](#),  
[PureNumber](#),  
[HexNumber](#),  
[IPv4Address](#),  
[MACAddress](#),  
[\\_UndefinedRepresentation](#) }
- recommended representation of a node value*
- enum [EEndianness](#) {  
[BigEndian](#),  
[LittleEndian](#),  
[\\_UndefinedEndian](#) }
- Endianness of a value in a register.*
- enum [ENameSpace](#) {  
[Custom](#),  
[Standard](#),  
[\\_UndefinedNameSpace](#) }
- Defines if a node name is standard or custom.*
- enum [EStandardNameSpace](#) {  
[None](#),  
[GEV](#),  
[IIDC](#),  
[CL](#),  
[USB](#),  
[\\_UndefinedStandardNameSpace](#) }
- Defines from which standard namespace a node name comes from.*
- enum [EYesNo](#) {  
[Yes](#) = 1,  
[No](#) = 0,  
[\\_UndefinedYesNo](#) = 2 }
- Defines the choices of a Yes/No alternative.*
- enum [ESlope](#) {  
[Increasing](#),  
[Decreasing](#),  
[Varying](#),  
[Automatic](#),  
[\\_UndefinedESlope](#) }
- typedef for formula type*
- enum [EXMLValidation](#) {  
[xvLoad](#) = 0x00000001L,  
[xvCycles](#) = 0x00000002L,  
[xvSFNC](#) = 0x00000004L,  
[xvDefault](#) = 0x00000000L,  
[xvAll](#) = 0xffffffffL,  
[\\_UndefinedEXMLValidation](#) = 0x80000000L }
- typedef describing the different validity checks which can be performed on an XML file*
- enum [EDisplayNotation](#) {  
[fnAutomatic](#),  
[fnFixed](#),  
[fnScientific](#),  
[\\_UndefinedEDisplayNotation](#) }
- typedef for float notation*

- enum `EInterfaceType` {  
`intfIValue`,  
`intfIBase`,  
`intfInteger`,  
`intfBoolean`,  
`intfCommand`,  
`intfFloat`,  
`intfString`,  
`intfRegister`,  
`intfCategory`,  
`intfEnumeration`,  
`intfEnumEntry`,  
`intfIPort` }

*typedef for interface type*

- enum `ELinkType` {  
`ctParentNodes`,  
`ctReadingChildren`,  
`ctWritingChildren`,  
`ctInvalidatingChildren`,  
`ctDependingNodes`,  
`ctTerminalNodes` }

*typedef for link type*

- enum `EIncMode` {  
`noIncrement`,  
`fixedIncrement`,  
`listIncrement` }

*typedef for increment mode*

- enum `EInputDirection` {  
`idFrom`,  
`idTo`,  
`idNone` }

*typedef for link type*

- enum `EGenApiSchemaVersion` {  
`v1_0` = 1,  
`v1_1` = 2,  
`_Undefined` = -1 }

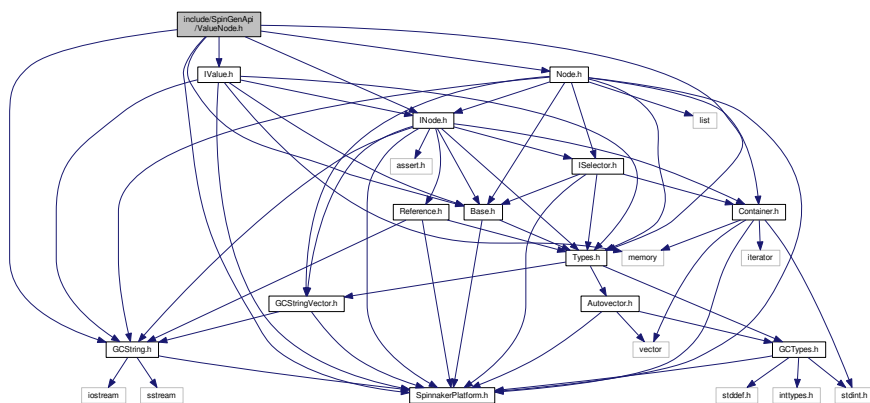
*GenApi schema version.*

## 11.121.1 Macro Definition Documentation

### 11.121.1.1 #define interface struct

## 11.122 include/SpinGenApi/ValueNode.h File Reference

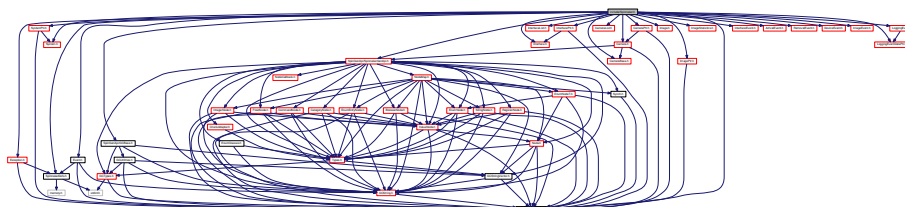
Include dependency graph for ValueNode.h:



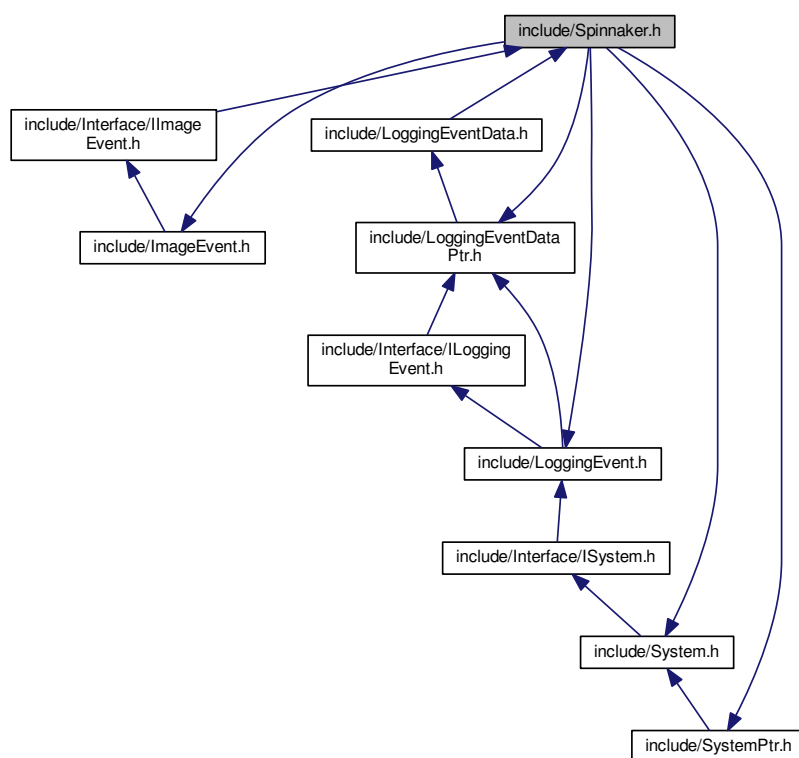


## 11.123 include/Spinnaker.h File Reference

Include dependency graph for Spinnaker.h:

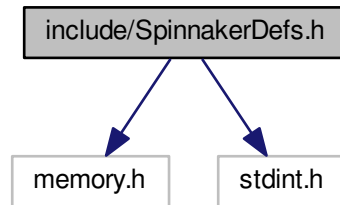


This graph shows which files directly or indirectly include this file:

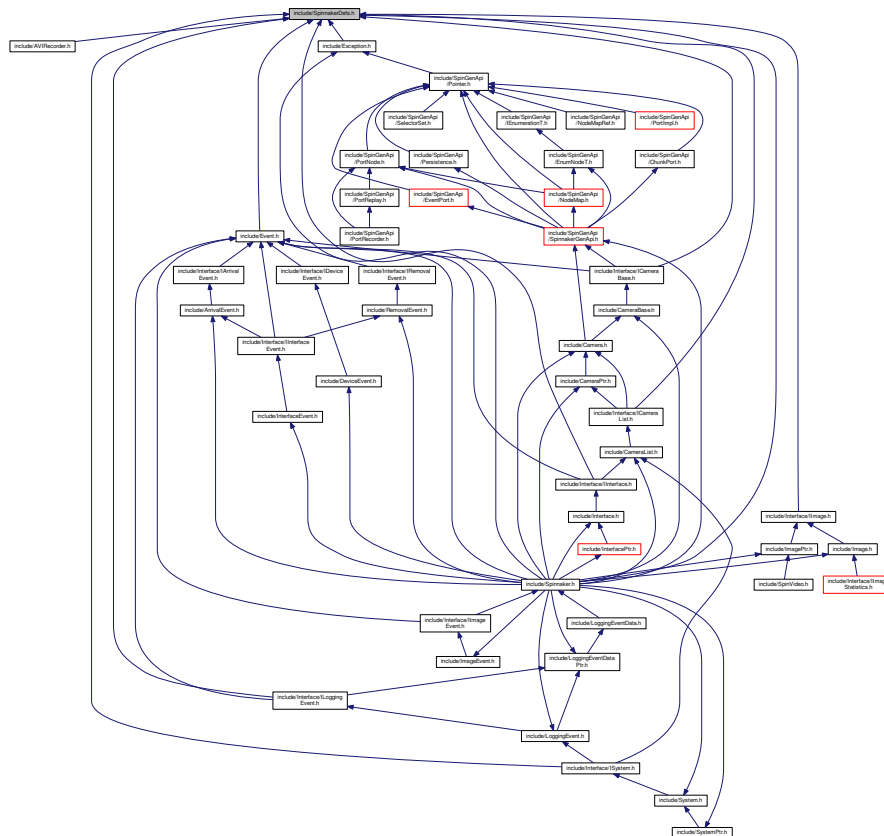


## 11.124 include/SpinnakerDefs.h File Reference

Include dependency graph for SpinnakerDefs.h:



This graph shows which files directly or indirectly include this file:



### Classes

- struct [PNGOption](#)  
Options for saving PNG images.

- struct [PPMOption](#)

*Options for saving PPM images.*

- struct [PGMOption](#)

*Options for saving PGM images.*

- struct [TIFFOption](#)

*Options for saving TIFF images.*

- struct [JPEGOption](#)

*Options for saving JPEG image.*

- struct [JPG2Option](#)

*Options for saving JPEG2000 image.*

- struct [BMPOption](#)

*Options for saving Bitmap image.*

- struct [LibraryVersion](#)

*Provides easier access to the current version of [Spinnaker](#).*

- struct [ActionCommandResult](#)

*Action Command Result.*

## Namespaces

- [Spinnaker](#)

## Enumerations

- enum `Error` {
  - `SPINNAKER_ERR_SUCCESS` = 0,
  - `SPINNAKER_ERR_ERROR` = -1001,
  - `SPINNAKER_ERR_NOT_INITIALIZED` = -1002,
  - `SPINNAKER_ERR_NOT_IMPLEMENTED` = -1003,
  - `SPINNAKER_ERR_RESOURCE_IN_USE` = -1004,
  - `SPINNAKER_ERR_ACCESS_DENIED` = -1005,
  - `SPINNAKER_ERR_INVALID_HANDLE` = -1006,
  - `SPINNAKER_ERR_INVALID_ID` = -1007,
  - `SPINNAKER_ERR_NO_DATA` = -1008,
  - `SPINNAKER_ERR_INVALID_PARAMETER` = -1009,
  - `SPINNAKER_ERR_IO` = -1010,
  - `SPINNAKER_ERR_TIMEOUT` = -1011,
  - `SPINNAKER_ERR_ABORT` = -1012,
  - `SPINNAKER_ERR_INVALID_BUFFER` = -1013,
  - `SPINNAKER_ERR_NOT_AVAILABLE` = -1014,
  - `SPINNAKER_ERR_INVALID_ADDRESS` = -1015,
  - `SPINNAKER_ERR_BUFFER_TOO_SMALL` = -1016,
  - `SPINNAKER_ERR_INVALID_INDEX` = -1017,
  - `SPINNAKER_ERR_PARSING_CHUNK_DATA` = -1018,
  - `SPINNAKER_ERR_INVALID_VALUE` = -1019,
  - `SPINNAKER_ERR_RESOURCE_EXHAUSTED` = -1020,
  - `SPINNAKER_ERR_OUT_OF_MEMORY` = -1021,
  - `SPINNAKER_ERR_BUSY` = -1022,
  - `GENICAM_ERR_INVALID_ARGUMENT` = -2001,
  - `GENICAM_ERR_OUT_OF_RANGE` = -2002,
  - `GENICAM_ERR_PROPERTY` = -2003,
  - `GENICAM_ERR_RUN_TIME` = -2004,
  - `GENICAM_ERR_LOGICAL` = -2005,
  - `GENICAM_ERR_ACCESS` = -2006,
  - `GENICAM_ERR_TIMEOUT` = -2007,
  - `GENICAM_ERR_DYNAMIC_CAST` = -2008,
  - `GENICAM_ERR_GENERIC` = -2009,
  - `GENICAM_ERR_BAD_ALLOCATION` = -2010,
  - `SPINNAKER_ERR_IM_CONVERT` = -3001,
  - `SPINNAKER_ERR_IM_COPY` = -3002,
  - `SPINNAKER_ERR_IM_MALLOC` = -3003,
  - `SPINNAKER_ERR_IM_NOT_SUPPORTED` = -3004,
  - `SPINNAKER_ERR_IM_HISTOGRAM_RANGE` = -3005,
  - `SPINNAKER_ERR_IM_HISTOGRAM_MEAN` = -3006,
  - `SPINNAKER_ERR_IM_MIN_MAX` = -3007,
  - `SPINNAKER_ERR_IM_COLOR_CONVERSION` = -3008,
  - `SPINNAKER_ERR_IM_DECOMPRESSION` = -3009,
  - `SPINNAKER_ERR_CUSTOM_ID` = -10000 }

*Spinnaker enum definitions.*

- enum `EventType` {
  - `SPINNAKER_EVENT_ARRIVAL_REMOVAL`,
  - `SPINNAKER_EVENT_DEVICE`,
  - `SPINNAKER_EVENT_DEVICE_SPECIFIC`,
  - `SPINNAKER_EVENT_NEW_BUFFER`,
  - `SPINNAKER_EVENT_LOGGING_EVENT`,
  - `SPINNAKER_EVENT_UNKNOWN` }

*Event types in Spinnaker.*

- enum `PixelFormatNamespaceID` {



```
SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,
SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,
SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,
SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }
```

*This enum represents the namespace in which the TL specific pixel format resides.*

- enum `ColorProcessingAlgorithm` {  
`DEFAULT`,  
`NO_COLOR_PROCESSING`,  
`NEAREST_NEIGHBOR`,  
`EDGE_SENSING`,  
`HQ_LINEAR`,  
`RIGOROUS`,  
`IPP`,  
`DIRECTIONAL_FILTER`,  
`WEIGHTED_DIRECTIONAL_FILTER` }

*Color processing algorithms.*

- enum `PolarizationAlgorithm` {  
`NO_POLARIZATION`,  
`QUADRANT_I0_GRAYSCALE`,  
`QUADRANT_I45_GRAYSCALE`,  
`QUADRANT_I90_GRAYSCALE`,  
`QUADRANT_I135_GRAYSCALE`,  
`STOKES_S0_GRAYSCALE`,  
`STOKES_S0_HEATMAP`,  
`STOKES_S1_GRAYSCALE`,  
`STOKES_S1_HEATMAP`,  
`STOKES_S2_GRAYSCALE`,  
`STOKES_S2_HEATMAP`,  
`DOLP_GRAYSCALE`,  
`DOLP_HEATMAP`,  
`AOP_GRAYSCALE`,  
`AOP_HEATMAP` }
- enum `PolarizationResolution` {  
`QUARTER_RESOLUTION`,  
`FULL_RESOLUTION` }
- enum `HeatMapColor` {  
`HEATMAP_BLACK` = 1,  
`HEATMAP_BLUE` = 2,  
`HEATMAP_CYAN` = 3,  
`HEATMAP_GREEN` = 4,  
`HEATMAP_YELLOW` = 5,  
`HEATMAP_RED` = 6,  
`HEATMAP_WHITE` = 7 }
- enum `ImageFileFormat` {  
`FROM_FILE_EXT` = -1,  
`PGM`,  
`PPM`,  
`BMP`,  
`JPEG`,  
`JPEG2000`,  
`TIFF`,  
`PNG`,  
`RAW`,  
`JPEG12_C`,  
`IMAGE_FILE_FORMAT_FORCE_32BITS` = 0x7FFFFFFF }

*File formats to be used for saving images to disk.*

- enum `ImageStatus` {  
`IMAGE_UNKNOWN_ERROR` = -1,  
`IMAGE_NO_ERROR` = 0,  
`IMAGE_CRC_CHECK_FAILED` = 1,  
`IMAGE_DATA_OVERFLOW` = 2,  
`IMAGE_MISSING_PACKETS` = 3,  
`IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT` = 4,  
`IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT` = 5,  
`IMAGE_PACKETID_INCONSISTENT` = 6,  
`IMAGE_MISSING_LEADER` = 7,  
`IMAGE_MISSING_TRAILER` = 8,  
`IMAGE_DATA_INCOMPLETE` = 9,  
`IMAGE_INFO_INCONSISTENT` = 10,  
`IMAGE_CHUNK_DATA_INVALID` = 11,  
`IMAGE_NO_SYSTEM_RESOURCES` = 12 }

*Status of images returned from `GetNextImage()` call.*

- enum `StatisticsChannel` {  
`GREY`,  
`RED`,  
`GREEN`,  
`BLUE`,  
`HUE`,  
`SATURATION`,  
`LIGHTNESS`,  
`NUM_STATISTICS_CHANNELS` }

*Channels that allow statistics to be calculated.*

- enum `SpinnakerLogLevel` {  
`LOG_LEVEL_OFF` = -1,  
`LOG_LEVEL_FATAL` = 0,  
`LOG_LEVEL_ALERT` = 100,  
`LOG_LEVEL_CRIT` = 200,  
`LOG_LEVEL_ERROR` = 300,  
`LOG_LEVEL_WARN` = 400,  
`LOG_LEVEL_NOTICE` = 500,  
`LOG_LEVEL_INFO` = 600,  
`LOG_LEVEL_DEBUG` = 700,  
`LOG_LEVEL_NOTSET` = 800 }

*log levels*

- enum `PayloadTypeInfoIDs` {  
`PAYLOAD_TYPE_UNKNOWN` = 0,  
`PAYLOAD_TYPE_IMAGE` = 1,  
`PAYLOAD_TYPE_RAW_DATA` = 2,  
`PAYLOAD_TYPE_FILE` = 3,  
`PAYLOAD_TYPE_CHUNK_DATA` = 4,  
`PAYLOAD_TYPE_JPEG` = 5,  
`PAYLOAD_TYPE_JPEG2000` = 6,  
`PAYLOAD_TYPE_H264` = 7,  
`PAYLOAD_TYPE_CHUNK_ONLY` = 8,  
`PAYLOAD_TYPE_DEVICE_SPECIFIC` = 9,  
`PAYLOAD_TYPE_MULTI_PART` = 10,  
`PAYLOAD_TYPE_CUSTOM_ID` = 1000,  
`PAYLOAD_TYPE_EXTENDED_CHUNK` = 1001 }
- enum `ActionCommandStatus` {  
`ACTION_COMMAND_STATUS_OK` = 0,  
`ACTION_COMMAND_STATUS_NO_REF_TIME` = 0x8013,  
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,  
`ACTION_COMMAND_STATUS_ACTION_LATE` = 0x8016,  
`ACTION_COMMAND_STATUS_ERROR` = 0x8FFF }

*Possible Status Codes Returned from Action Command.*

- enum [PixelFormatIntType](#) {  
[IntType\\_UINT8](#),  
[IntType\\_INT8](#),  
[IntType\\_UINT10](#),  
[IntType\\_UINT10p](#),  
[IntType\\_UINT10P](#),  
[IntType\\_UINT12](#),  
[IntType\\_UINT12p](#),  
[IntType\\_UINT12P](#),  
[IntType\\_UINT14](#),  
[IntType\\_UINT16](#),  
[IntType\\_FLOAT32](#),  
[IntType\\_UNKNOWN](#) }

*Possible integer types and packing used in a pixel format.*

## Variables

- const uint64\_t [EVENT\\_TIMEOUT\\_NONE](#) = 0  
*Timeout values for getting next image, device, or interface event.*
- const uint64\_t [EVENT\\_TIMEOUT\\_INFINITE](#) = 0xFFFFFFFFFFFFFFFF

## 11.125 include/SpinnakerPlatform.h File Reference

### Macros

- #define [SPINNAKER\\_API\\_ABSTRACT](#) /\*nothing\*/
- #define [SPINNAKER\\_API](#) \_\_attribute\_\_((visibility ("default"))) )
- #define [SPINNAKER\\_LOCAL](#) \_\_attribute\_\_((visibility ("hidden"))) )

## 11.126 include/SpinUpdate.h File Reference

### Macros

- #define [SPINUPDATE\\_API](#) SPINUPDATE\_IMPORT\_EXPORT

### Functions

- [SPINUPDATE\\_API](#) int [UpdateFirmwareConsole](#) (int argc, char \*\*argv)  
*Updates the firmware for the device.*
- [SPINUPDATE\\_API](#) int [UpdateFirmware](#) (const char \*args)
- [SPINUPDATE\\_API](#) void [SetMessageCallback](#) ([UpdaterMessageCallback](#) messageCallbackFunction)
- [SPINUPDATE\\_API](#) void [SetProgressCallback](#) ([UpdaterProgressCallback](#) progressCallbackFunction)
- [SPINUPDATE\\_API](#) const char \* [GetErrorMessage](#) ()

## Variables

- `SPINUPDATE_API` typedef int(\* [UpdaterMessageCallback](#) )(const char \*message)
- `SPINUPDATE_API` typedef int(\* [UpdaterProgressCallback](#) )(const char \*action, unsigned int address, int globalPercent, int currPercent)

## 11.126.1 Macro Definition Documentation

11.126.1.1 `#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT`

## 11.126.2 Function Documentation

11.126.2.1 `SPINUPDATE_API const char* GetErrorMessage ( )`

11.126.2.2 `SPINUPDATE_API void SetMessageCallback ( UpdaterMessageCallback messageCallbackFunction )`

11.126.2.3 `SPINUPDATE_API void SetProgressCallback ( UpdaterProgressCallback progressCallbackFunction )`

11.126.2.4 `SPINUPDATE_API int UpdateFirmware ( const char * args )`

11.126.2.5 `SPINUPDATE_API int UpdateFirmwareConsole ( int argc, char ** argv )`

Updates the firmware for the device.

### Parameters

|             |                                                           |
|-------------|-----------------------------------------------------------|
| <i>argc</i> | Number of strings pointed to by argv                      |
| <i>argv</i> | Pointer to list of string options for the firmware update |

### Returns

0 for success, otherwise non zero for failures.

Typical usage for updating is as follows: `-R{serial number} [-{options} ..] {firmware zim file} -R{serial number} -UU -B {firmware zim file}`

Option definitions: `-B` = Reboots the camera after the update has completed. If this argument is not provided, a manual power cycle will be required. `-A` = Updates individual portions of the firmware in flash. The code section of camera at location `0xFF08000` will be updated. `./sample_app -AFF08000 camera.zim` `-U` = Downgrade the firmware. Multiple U's can be used to overwrite the ROM header. `-F` = Force program and EEPROM reload. `-R` = Enter a regular expression for camera serial match. For example: `./sample_app -R.* camera.zim` Results in matching any camera serial `-P` = Checks the progress of the updater. `-epromsave` = Save the content of the EEPROM to a file.

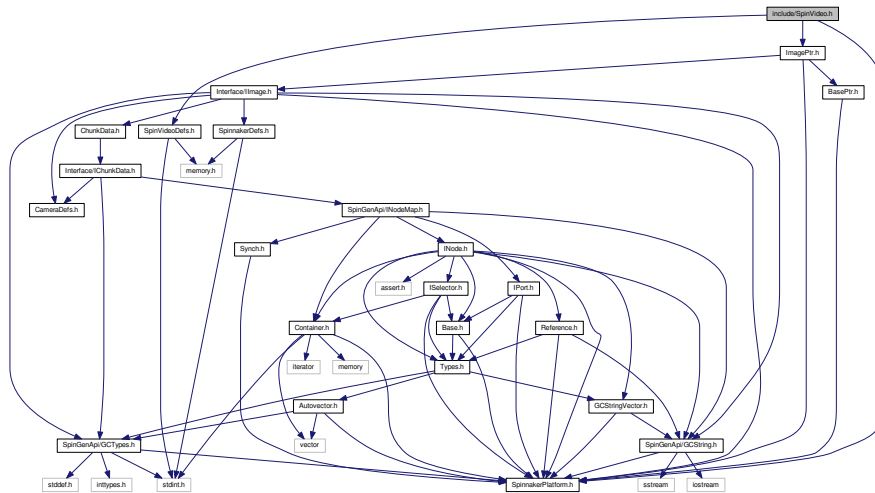
## 11.126.3 Variable Documentation

11.126.3.1 `SPINUPDATE_API typedef int(* UpdaterMessageCallback )(const char *message)`

11.126.3.2 **SPINUPDATE\_API** typedef int(\* UpdatorProgressCallback) (const char \*action, unsigned int address, int globalPercent, int currPercent)

## 11.127 include/SpinVideo.h File Reference

Include dependency graph for SpinVideo.h:



### Classes

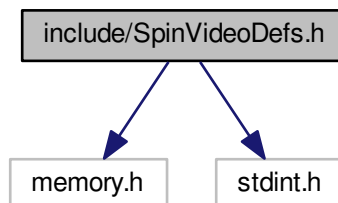
- class [SpinVideo](#)  
*Provides the functionality for the user to record images to an AVI/MP4 file.*

### Namespaces

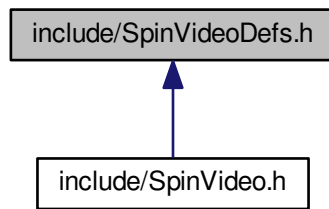
- [Spinnaker](#)
- [Spinnaker::Video](#)

## 11.128 include/SpinVideoDefs.h File Reference

Include dependency graph for SpinVideoDefs.h:



This graph shows which files directly or indirectly include this file:



## Classes

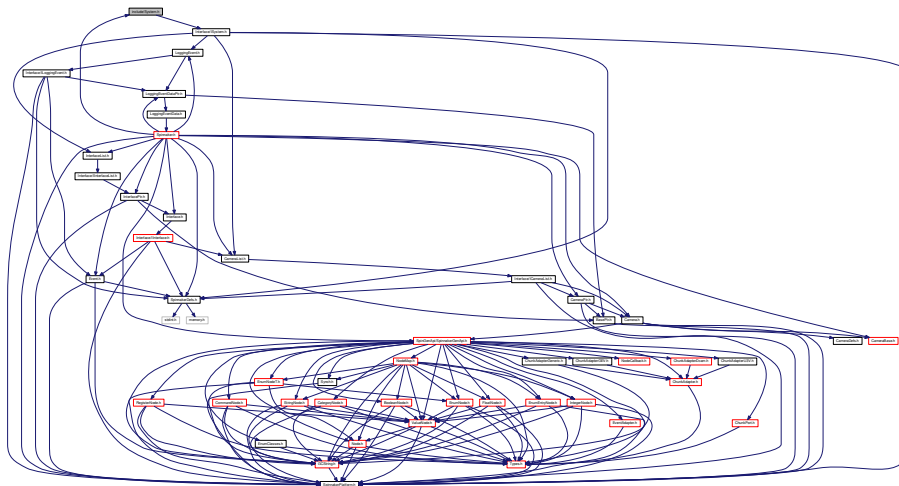
- struct [MJPGOption](#)  
*Options for saving MJPG files.*
- struct [H264Option](#)  
*Options for saving H264 files.*
- struct [AVIOption](#)  
*Options for saving AVI files.*

## Namespaces

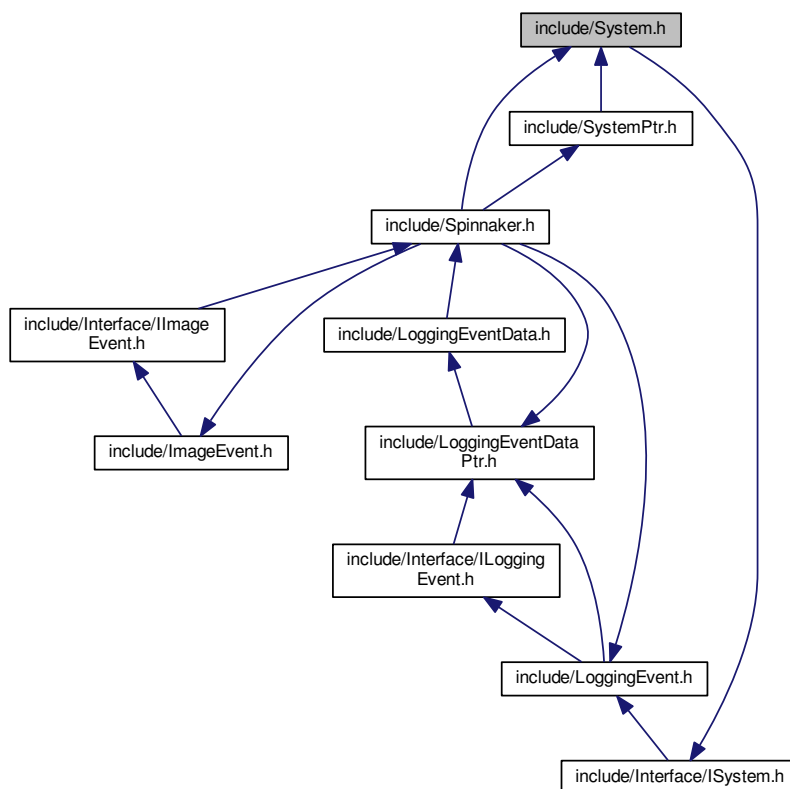
- [Spinnaker](#)
- [Spinnaker::Video](#)

## 11.129 include/System.h File Reference

Include dependency graph for System.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [System](#)

*The system object is used to retrieve the list of interfaces and cameras available.*

## Namespaces

- [Spinnaker](#)

## Macros

- `#define FLIR_SPINNAKER_VERSION_MAJOR 1`
- `#define FLIR_SPINNAKER_VERSION_MINOR 18`
- `#define FLIR_SPINNAKER_VERSION_TYPE 0`
- `#define FLIR_SPINNAKER_VERSION_BUILD 17`

### 11.129.1 Macro Definition Documentation

11.129.1.1 `#define FLIR_SPINNAKER_VERSION_BUILD 17`

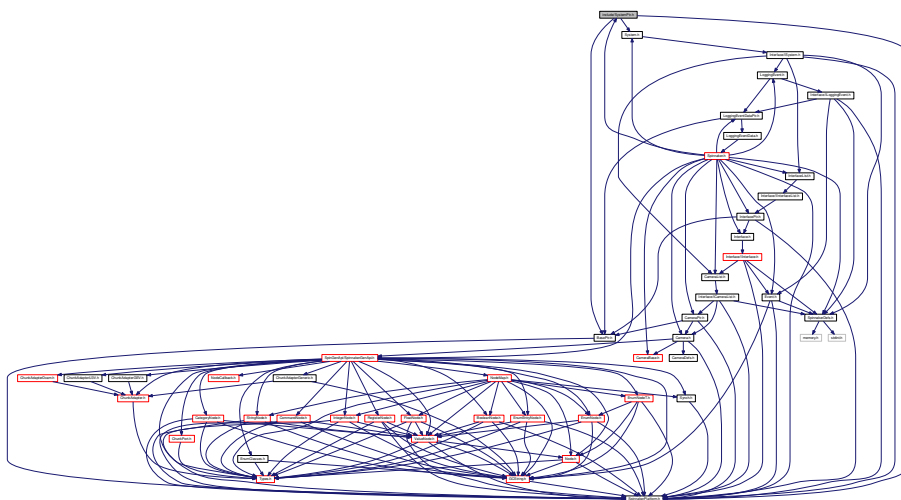
11.129.1.2 `#define FLIR_SPINNAKER_VERSION_MAJOR 1`

11.129.1.3 `#define FLIR_SPINNAKER_VERSION_MINOR 18`

11.129.1.4 `#define FLIR_SPINNAKER_VERSION_TYPE 0`

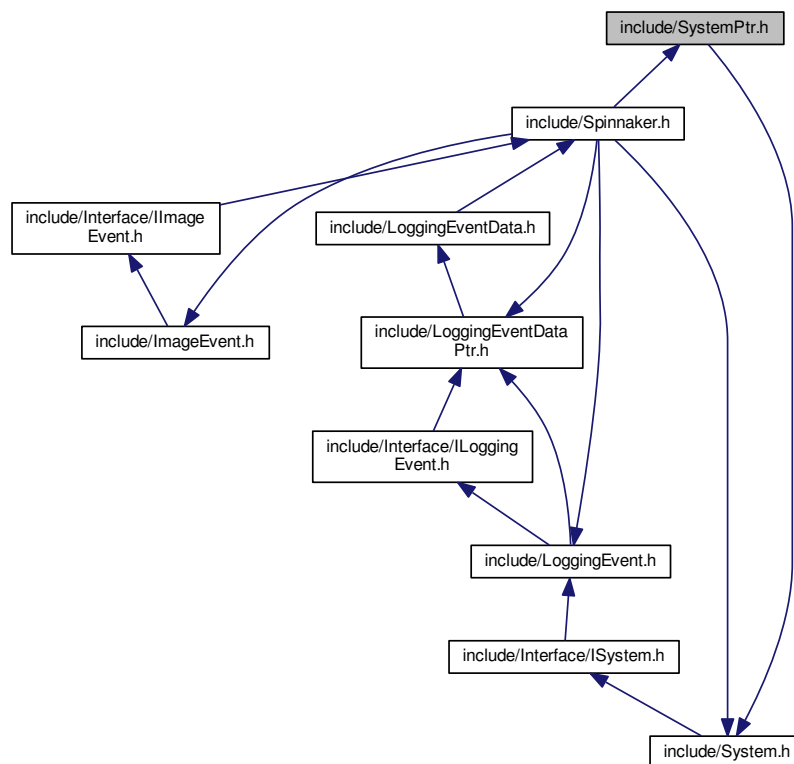
### 11.130 include/SystemPtr.h File Reference

Include dependency graph for SystemPtr.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [SystemPtr](#)

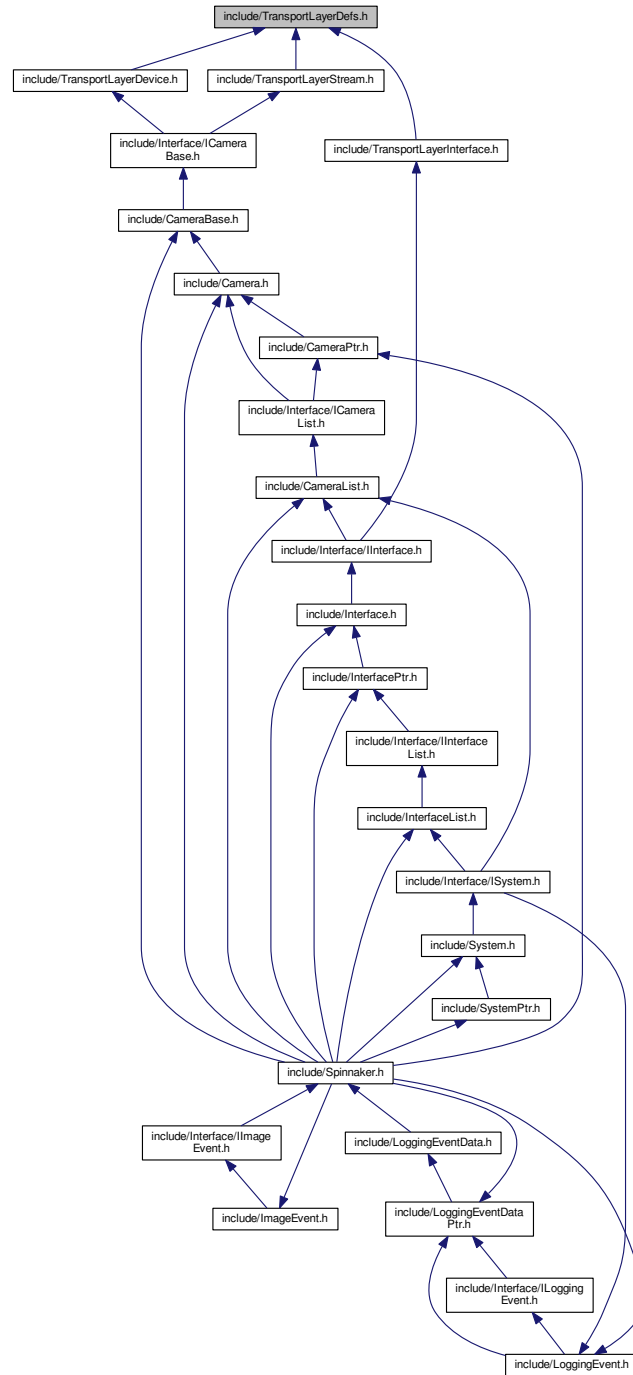
*A reference tracked pointer to a system object.*

## Namespaces

- [Spinnaker](#)

## 11.131 include/TransportLayerDefs.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Spinnaker](#)

## Enumerations

- enum [StreamTypeEnum](#) {  
[StreamType\\_Mixed](#),  
[StreamType\\_Custom](#),  
[StreamType\\_GEV](#),  
[StreamType\\_CL](#),  
[StreamType\\_IIDC](#),  
[StreamType\\_UVC](#),  
[StreamType\\_CXP](#),  
[StreamType\\_CLHS](#),  
[StreamType\\_U3V](#),  
[StreamType\\_ETHERNET](#),  
[StreamType\\_PCI](#),  
[NUMSTREAMTYPE](#) }

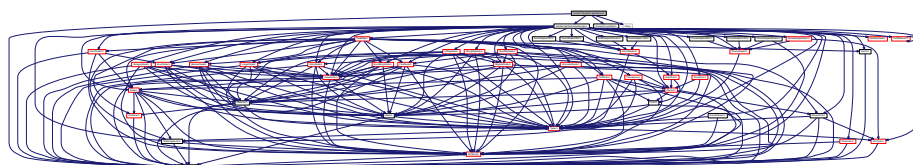
*The enum definitions for TL Device nodes from the transport layer .xml files.*

- enum [StreamDefaultBufferCountModeEnum](#) {  
[StreamDefaultBufferCountMode\\_Manual](#),  
[StreamDefaultBufferCountMode\\_Auto](#),  
[NUMSTREAMDEFAULTBUFFERCOUNTMODE](#) }
- enum [StreamBufferCountModeEnum](#) {  
[StreamBufferCountMode\\_Manual](#),  
[StreamBufferCountMode\\_Auto](#),  
[NUMSTREAMBUFFERCOUNTMODE](#) }
- enum [StreamBufferHandlingModeEnum](#) {  
[StreamBufferHandlingMode\\_OldestFirst](#),  
[StreamBufferHandlingMode\\_OldestFirstOverwrite](#),  
[StreamBufferHandlingMode\\_NewestFirst](#),  
[StreamBufferHandlingMode\\_NewestFirstOverwrite](#),  
[StreamBufferHandlingMode\\_NewestOnly](#),  
[NUMSTREAMBUFFERHANDLINGMODE](#) }
- enum [DeviceTypeEnum](#) {  
[DeviceType\\_Mixed](#),  
[DeviceType\\_Custom](#),  
[DeviceType\\_GEV](#),  
[DeviceType\\_CL](#),  
[DeviceType\\_IIDC](#),  
[DeviceType\\_UVC](#),  
[DeviceType\\_CXP](#),  
[DeviceType\\_CLHS](#),  
[DeviceType\\_U3V](#),  
[DeviceType\\_ETHERNET](#),  
[DeviceType\\_PCI](#),  
[NUMDEVICETYPE](#) }
- enum [DeviceAccessStatusEnum](#) {  
[DeviceAccessStatus\\_Unknown](#),  
[DeviceAccessStatus\\_ReadWrite](#),  
[DeviceAccessStatus\\_ReadOnly](#),  
[DeviceAccessStatus\\_NoAccess](#),  
[NUMDEVICEACCESSSTATUS](#) }
- enum [GevCCPEnum](#) {  
[GevCCP\\_EnumEntry\\_GevCCP\\_OpenAccess](#),  
[GevCCP\\_EnumEntry\\_GevCCP\\_ExclusiveAccess](#),  
[GevCCP\\_EnumEntry\\_GevCCP\\_ControlAccess](#),  
[NUMGEVCCP](#) }
- enum [GUIXMLLocationEnum](#) {  
[GUIXMLLocation\\_Device](#),  
[GUIXMLLocation\\_Host](#),  
[NUMGUIXMLLOCATION](#) }

- enum [GenICamXMLLocationEnum](#) {  
[GenICamXMLLocation\\_Device](#),  
[GenICamXMLLocation\\_Host](#),  
[NUMGENICAMXMLLOCATION](#) }
- enum [DeviceEndiannessMechanismEnum](#) {  
[DeviceEndiannessMechanism\\_Legacy](#),  
[DeviceEndiannessMechanism\\_Standard](#),  
[NUMDEVICEENDIANESSMECHANISM](#) }
- enum [DeviceCurrentSpeedEnum](#) {  
[DeviceCurrentSpeed\\_UnknownSpeed](#),  
[DeviceCurrentSpeed\\_LowSpeed](#),  
[DeviceCurrentSpeed\\_FullSpeed](#),  
[DeviceCurrentSpeed\\_HighSpeed](#),  
[DeviceCurrentSpeed\\_SuperSpeed](#),  
[NUMDEVICECURRENTSPEED](#) }
- enum [POEStatusEnum](#) {  
[POEStatus\\_NotSupported](#),  
[POEStatus\\_PowerOff](#),  
[POEStatus\\_PowerOn](#),  
[NUMPOESTATUS](#) }

## 11.132 [include/TransportLayerDevice.h](#) File Reference

Include dependency graph for [TransportLayerDevice.h](#):



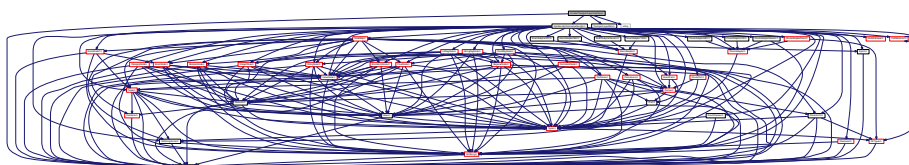
- class `TransportLayerDevice`

## Namespaces

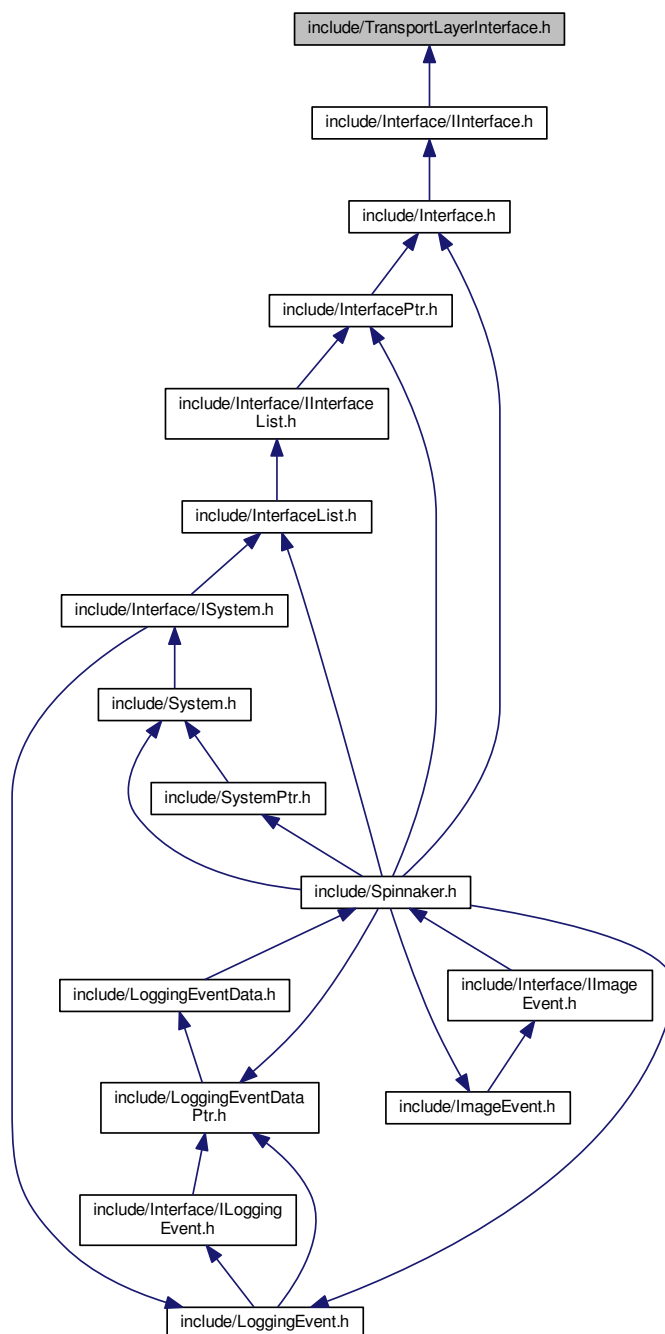
- Spinnaker

### 11.133 include/TransportLayerInterface.h File Reference

Include dependency graph for TransportLayerInterface.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [TransportLayerInterface](#)

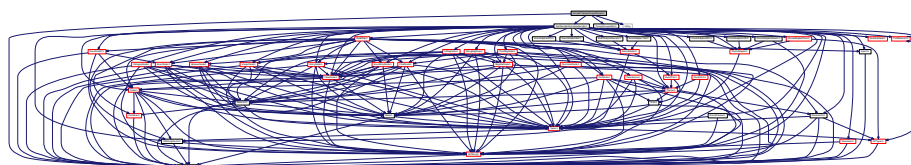
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

- [Spinnaker](#)

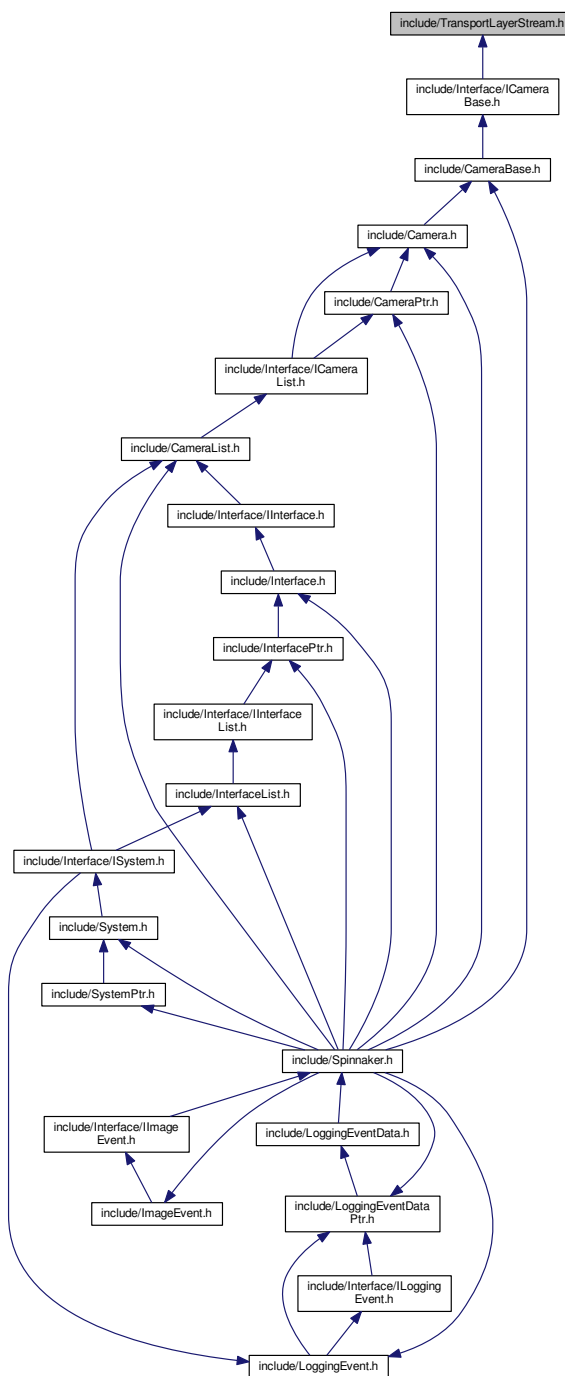
## 11.134 include/TransportLayerStream.h File Reference

Include dependency graph for TransportLayerStream.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class `TransportLayerStream`

*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

- Spinnaker



# Index

- [\\_ClearXMLCache](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [593](#)
  - [\\_Connect](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [593](#)
  - [\\_CycleDetectAccessMode](#)
    - Types Enums, [311](#)
  - [\\_Destroy](#)
    - Spinnaker GenApi Classes, [196](#)
  - [\\_GetDeviceName](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_GetNode](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_GetNodes](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_GetSupportedSchemaVersions](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_Initialize](#)
    - Spinnaker::GenApi::CGeneric\_XMLLoader↔  
Params, [562](#)
  - [\\_InvalidateNodes](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_LoadXMLFromFile](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_LoadXMLFromFileInject](#)
    - Spinnaker GenApi Classes, [196](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_LoadXMLFromString](#)
    - Spinnaker GenApi Classes, [197](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_LoadXMLFromStringInject](#)
    - Spinnaker GenApi Classes, [197](#)
    - Spinnaker::GenApi::CNodeMapRefT, [594](#)
  - [\\_LoadXMLFromZIPData](#)
    - Spinnaker GenApi Classes, [197](#)
    - Spinnaker::GenApi::CNodeMapRefT, [595](#)
  - [\\_LoadXMLFromZIPFile](#)
    - Spinnaker GenApi Classes, [197](#)
    - Spinnaker::GenApi::CNodeMapRefT, [595](#)
  - [\\_Poll](#)
    - Spinnaker GenApi Classes, [197](#)
    - Spinnaker::GenApi::CNodeMapRefT, [595](#)
- [\\_Ptr](#)
    - Spinnaker::GenApi::CNodeMapRefT, [595](#)
    - Spinnaker::GenApi::NodeMap, [784](#)
  - [\\_TO\\_STRING](#)
    - GCUtilities.h, [1004](#)
  - [\\_UndefinedRepresentation](#)
    - Types Enums, [310](#)
  - [\\_Undefined](#)
    - Types Enums, [312](#)
  - [\\_UndefinedAccessMode](#)
    - Types Enums, [311](#)
  - [\\_UndefinedCachingMode](#)
    - Types Enums, [311](#)
  - [\\_UndefinedEDisplayNotation](#)
    - Types Enums, [311](#)
  - [\\_UndefinedESlope](#)
    - Types Enums, [314](#)
  - [\\_UndefinedEXMLValidation](#)
    - Types Enums, [314](#)
  - [\\_UndefinedEndian](#)
    - Types Enums, [311](#)
  - [\\_UndefinedNameSpace](#)
    - Types Enums, [313](#)
  - [\\_UndefinedRepresentation](#)
    - Types Enums, [313](#)
  - [\\_UndefinedSign](#)
    - Types Enums, [313](#)
  - [\\_UndefinedStandardNameSpace](#)
    - Types Enums, [314](#)
  - [\\_UndefinedVisibility](#)
    - Types Enums, [314](#)
  - [\\_UndefinedYesNo](#)
    - Types Enums, [315](#)
  - [\\_\\_ERR\\_\\_](#)
    - GCUtilities.h, [1004](#)
  - [\\_\\_LINE\\_STR\\_\\_](#)
    - GCUtilities.h, [1004](#)
  - [\\_\\_LOCATION\\_\\_](#)
    - GCUtilities.h, [1004](#)
  - [\\_\\_OUTPUT\\_FORMATER\\_\\_](#)
    - GCUtilities.h, [1004](#)
  - [\\_\\_STDC\\_CONSTANT\\_MACROS](#)
    - GCTypes.h, [1000](#)
  - [\\_\\_STDC\\_LIMIT\\_MACROS](#)
    - GCTypes.h, [1000](#)
  - [\\_\\_TODO\\_\\_](#)
    - GCUtilities.h, [1004](#)
  - [\\_\\_WARN\\_\\_](#)
    - GCUtilities.h, [1004](#)

- `_npos`
  - `Spinnaker::GenICam::gcstring`, 655
- `_pCount`
  - `Spinnaker::GenApi::double_autovector_t`, 619
  - `Spinnaker::GenApi::int64_autovector_t`, 734
- `_pv`
  - `Spinnaker::GenApi::double_autovector_t`, 619
  - `Spinnaker::GenApi::int64_autovector_t`, 734
- `~ArrivalEvent`
  - `Spinnaker::ArrivalEvent`, 377
- `~AutoLock`
  - `Spinnaker::GenApi::AutoLock`, 378
  - `Spinnaker::GenICam::AutoLock`, 379
- `~BasePtr`
  - `Spinnaker::BasePtr`, 381
- `~BooleanNode`
  - `Spinnaker::GenApi::BooleanNode`, 385
- `~CChunkAdapter`
  - `Spinnaker::GenApi::CChunkAdapter`, 530
- `~CChunkAdapterDcam`
  - `Spinnaker::GenApi::CChunkAdapterDcam`, 532
- `~CChunkAdapterGEV`
  - `Spinnaker::GenApi::CChunkAdapterGEV`, 536
- `~CChunkAdapterGeneric`
  - `Spinnaker::GenApi::CChunkAdapterGeneric`, 534
- `~CChunkAdapterU3V`
  - `Spinnaker::GenApi::CChunkAdapterU3V`, 538
- `~CChunkPort`
  - `Spinnaker::GenApi::CChunkPort`, 540
- `~CEnumerationTRef`
  - `Spinnaker::GenApi::CEnumerationTRef`, 544
- `~CEventAdapter`
  - `Spinnaker::GenApi::CEventAdapter`, 547
- `~CEventAdapter1394`
  - `Spinnaker::GenApi::CEventAdapter1394`, 549
- `~CEventAdapterGEV`
  - `Spinnaker::GenApi::CEventAdapterGEV`, 552
- `~CEventAdapterGeneric`
  - `Spinnaker::GenApi::CEventAdapterGeneric`, 550
- `~CEventAdapterU3V`
  - `Spinnaker::GenApi::CEventAdapterU3V`, 554
- `~CEventPort`
  - `Spinnaker::GenApi::CEventPort`, 556
- `~CFeatureBag`
  - `Spinnaker::GenApi::CFeatureBag`, 559
- `~CGlobalLock`
  - `Spinnaker::GenICam::CGlobalLock`, 563
- `~CGlobalLockUnlocker`
  - `Spinnaker::GenICam::CGlobalLockUnlocker`, 565
- `~CLock`
  - `Spinnaker::GenApi::CLock`, 576
  - `Spinnaker::GenICam::CLock`, 577
- `~CNodeCallback`
  - `Spinnaker::GenApi::CNodeCallback`, 581
- `~CNodeMapFactory`
  - `Spinnaker::GenApi::CNodeMapFactory`, 584
- `~CNodeMapRefT`
  - `Spinnaker GenApi Classes`, 198
- `~CPointer`
  - `Spinnaker::GenApi::CPointer`, 600
- `~CPortImpl`
  - `Spinnaker::GenApi::CPortImpl`, 603
- `~CPortWriteList`
  - `Spinnaker::GenApi::CPortWriteList`, 606
- `~CRegisterPortImpl`
  - `Spinnaker::GenApi::CRegisterPortImpl`, 608
- `~CSelectorSet`
  - `Spinnaker::GenApi::CSelectorSet`, 611
- `~Camera`
  - `Spinnaker::Camera`, 416
- `~CameraBase`
  - `Spinnaker::CameraBase`, 515
- `~CameraList`
  - `Spinnaker::CameraList`, 523
- `~CameraPtr`
  - `CameraPtr Class`, 137
- `~CategoryNode`
  - `Spinnaker::GenApi::CategoryNode`, 528
- `~ChunkData`
  - `Spinnaker::ChunkData`, 568
- `~CommandNode`
  - `Spinnaker::GenApi::CommandNode`, 597
- `~DeviceEvent`
  - `Spinnaker::DeviceEvent`, 616
- `~EnumEntryNode`
  - `Spinnaker::GenApi::EnumEntryNode`, 627
- `~EnumNode`
  - `Spinnaker::GenApi::EnumNode`, 630
- `~Event`
  - `Spinnaker::Event`, 636
- `~Exception`
  - `Spinnaker::Exception`, 640
- `~FileProtocolAdapter`
  - `Spinnaker::GenApi::FileProtocolAdapter`, 643
- `~FloatNode`
  - `Spinnaker::GenApi::FloatNode`, 647
- `~FloatRegNode`
  - `Spinnaker::GenApi::FloatRegNode`, 651
- `~IArrivalEvent`
  - `Spinnaker::IArrivalEvent`, 667
- `~ICameraBase`
  - `Spinnaker::ICameraBase`, 669
- `~ICameraList`
  - `Spinnaker::ICameraList`, 673
- `~IChunkData`
  - `Spinnaker::IChunkData`, 676
- `~IDevFileStreamBuf`
  - `Spinnaker::GenApi::IDevFileStreamBuf`, 683
- `~IDeviceEvent`
  - `Spinnaker::IDeviceEvent`, 684
- `~IImage`
  - `Spinnaker::IImage`, 687
- `~IImageEvent`
  - `Spinnaker::IImageEvent`, 693
- `~IImageStatistics`
  - `Spinnaker::IImageStatistics`, 694

- ~IInterface
  - Spinnaker::IInterface, [697](#)
- ~IInterfaceEvent
  - Spinnaker::IInterfaceEvent, [700](#)
- ~IInterfaceList
  - Spinnaker::IInterfaceList, [701](#)
- ~ILoggingEvent
  - Spinnaker::ILoggingEvent, [703](#)
- ~IRemovalEvent
  - Spinnaker::IRemovalEvent, [753](#)
- ~ISystem
  - Spinnaker::ISystem, [754](#)
- ~Image
  - Spinnaker::Image, [707](#)
- ~ImageEvent
  - Spinnaker::ImageEvent, [725](#)
- ~ImagePtr
  - Spinnaker::ImagePtr, [727](#)
- ~ImageStatistics
  - Spinnaker::ImageStatistics, [729](#)
- ~IntRegNode
  - Spinnaker::GenApi::IntRegNode, [751](#)
- ~IntegerNode
  - Spinnaker::GenApi::IntegerNode, [737](#)
- ~Interface
  - Spinnaker::Interface, [741](#)
- ~InterfaceEvent
  - Spinnaker::InterfaceEvent, [745](#)
- ~InterfaceList
  - Spinnaker::InterfaceList, [747](#)
- ~InterfacePtr
  - Spinnaker::InterfacePtr, [749](#)
- ~Lock
  - Spinnaker::GenICam::LockableObject::Lock, [759](#)
- ~LoggingEvent
  - Spinnaker::LoggingEvent, [762](#)
- ~LoggingEventData
  - Spinnaker::LoggingEventData, [764](#)
- ~LoggingEventDataPtr
  - Spinnaker::LoggingEventDataPtr, [767](#)
- ~Node
  - Spinnaker::GenApi::Node, [773](#)
- ~NodeMap
  - Spinnaker::GenApi::NodeMap, [780](#)
- ~ODevFileStreamBuf
  - Spinnaker::GenApi::ODevFileStreamBuf, [788](#)
- ~PortNode
  - Spinnaker::GenApi::PortNode, [792](#)
- ~PortRecorder
  - Spinnaker::GenApi::PortRecorder, [796](#)
- ~PortReplay
  - Spinnaker::GenApi::PortReplay, [798](#)
- ~RegisterNode
  - Spinnaker::GenApi::RegisterNode, [802](#)
- ~RemovalEvent
  - Spinnaker::RemovalEvent, [804](#)
- ~SpinVideo
  - Spinnaker::Video::SpinVideo, [807](#)
- ~StringNode
  - Spinnaker::GenApi::StringNode, [811](#)
- ~StringRegNode
  - Spinnaker::GenApi::StringRegNode, [814](#)
- ~System
  - Spinnaker::System, [817](#)
- ~SystemPtr
  - Spinnaker::SystemPtr, [823](#)
- ~TransportLayerDevice
  - Spinnaker::TransportLayerDevice, [827](#)
- ~TransportLayerInterface
  - Spinnaker::TransportLayerInterface, [834](#)
- ~TransportLayerStream
  - Spinnaker::TransportLayerStream, [841](#)
- ~ValueNode
  - Spinnaker::GenApi::ValueNode, [848](#)
- ~double\_autovector\_t
  - Spinnaker::GenApi::double\_autovector\_t, [618](#)
- ~gcstring
  - Spinnaker::GenICam::gcstring, [655](#)
- ~int64\_autovector\_t
  - Spinnaker::GenApi::int64\_autovector\_t, [734](#)
- ACTION\_COMMAND\_STATUS\_ACTION\_LATE
  - Spinnaker Definitions, [161](#)
- ACTION\_COMMAND\_STATUS\_ERROR
  - Spinnaker Definitions, [161](#)
- ACTION\_COMMAND\_STATUS\_NO\_REF\_TIME
  - Spinnaker Definitions, [161](#)
- ACTION\_COMMAND\_STATUS\_OVERFLOW
  - Spinnaker Definitions, [161](#)
- ACTION\_COMMAND\_STATUS\_OK
  - Spinnaker Definitions, [161](#)
- ADOBE\_DEFLATE
  - Spinnaker::TIFFOption, [824](#)
- AOP\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- AOP\_HEATMAP
  - Spinnaker Definitions, [166](#)
- aPAUSEMACCtrlFramesReceived
  - Spinnaker::Camera, [420](#)
- aPAUSEMACCtrlFramesTransmitted
  - Spinnaker::Camera, [420](#)
- AVI Recorder Class, [33](#)
  - DEPRECATED\_CLASS, [33](#)
- AVIOption, [379](#)
  - Spinnaker::Video::AVIOption, [379](#)
- AasRoiEnable
  - Spinnaker::Camera, [416](#)
- AasRoiHeight
  - Spinnaker::Camera, [416](#)
- AasRoiOffsetX
  - Spinnaker::Camera, [416](#)
- AasRoiOffsetY
  - Spinnaker::Camera, [416](#)
- AasRoiWidth
  - Spinnaker::Camera, [417](#)
- AcquisitionAbort
  - Spinnaker::Camera, [417](#)

- AcquisitionArm
  - Spinnaker::Camera, [417](#)
- AcquisitionBurstFrameCount
  - Spinnaker::Camera, [417](#)
- AcquisitionFrameCount
  - Spinnaker::Camera, [417](#)
- AcquisitionFrameRate
  - Spinnaker::Camera, [418](#)
- AcquisitionFrameRateEnable
  - Spinnaker::Camera, [418](#)
- AcquisitionLineRate
  - Spinnaker::Camera, [418](#)
- AcquisitionMode
  - Spinnaker::Camera, [418](#)
- AcquisitionMode\_Continuous
  - CameraDefs Class, [68](#)
- AcquisitionMode\_MultiFrame
  - CameraDefs Class, [68](#)
- AcquisitionMode\_SingleFrame
  - CameraDefs Class, [68](#)
- AcquisitionModeEnums
  - CameraDefs Class, [68](#)
- AcquisitionResultingFrameRate
  - Spinnaker::Camera, [418](#)
- AcquisitionStart
  - Spinnaker::Camera, [418](#)
- AcquisitionStatus
  - Spinnaker::Camera, [418](#)
- AcquisitionStatusSelector
  - Spinnaker::Camera, [419](#)
- AcquisitionStatusSelector\_AcquisitionActive
  - CameraDefs Class, [69](#)
- AcquisitionStatusSelector\_AcquisitionTransfer
  - CameraDefs Class, [69](#)
- AcquisitionStatusSelector\_AcquisitionTriggerWait
  - CameraDefs Class, [69](#)
- AcquisitionStatusSelector\_ExposureActive
  - CameraDefs Class, [69](#)
- AcquisitionStatusSelector\_FrameActive
  - CameraDefs Class, [69](#)
- AcquisitionStatusSelector\_FrameTriggerWait
  - CameraDefs Class, [69](#)
- AcquisitionStatusSelectorEnums
  - CameraDefs Class, [68](#)
- AcquisitionStop
  - Spinnaker::Camera, [419](#)
- ActionCommand
  - Spinnaker::TransportLayerInterface, [834](#)
- ActionCommandResult, [375](#)
- ActionCommandStatus
  - Spinnaker Definitions, [161](#)
- ActionDeviceKey
  - Spinnaker::Camera, [419](#)
- ActionGroupKey
  - Spinnaker::Camera, [419](#)
- ActionGroupMask
  - Spinnaker::Camera, [419](#)
- ActionQueueSize
  - Spinnaker::Camera, [419](#)
- ActionSelector
  - Spinnaker::Camera, [419](#)
- ActionUnconditionalMode
  - Spinnaker::Camera, [420](#)
- ActionUnconditionalMode\_Off
  - CameraDefs Class, [69](#)
- ActionUnconditionalMode\_On
  - CameraDefs Class, [69](#)
- ActionUnconditionalModeEnums
  - CameraDefs Class, [69](#)
- AdaptiveCompressionEnable
  - Spinnaker::Camera, [420](#)
- AdcBitDepth
  - Spinnaker::Camera, [420](#)
- AdcBitDepth\_Bit10
  - CameraDefs Class, [69](#)
- AdcBitDepth\_Bit12
  - CameraDefs Class, [69](#)
- AdcBitDepth\_Bit14
  - CameraDefs Class, [69](#)
- AdcBitDepth\_Bit8
  - CameraDefs Class, [69](#)
- AdcBitDepthEnums
  - CameraDefs Class, [69](#)
- AddInjectionData
  - Spinnaker::GenApi::CNodeMapFactory, [586](#)
- Address
  - IPort Interface, [268](#)
- Append
  - Spinnaker::CameraList, [523](#)
  - Spinnaker::ICameraList, [673](#)
  - Spinnaker::Video::SpinVideo, [807](#)
- append
  - Spinnaker::GenICam::gcstring, [655](#)
- ApplyStyleSheet
  - Spinnaker::GenApi::CNodeMapFactory, [586](#)
- ArrivalEvent, [376](#)
  - Spinnaker::ArrivalEvent, [377](#)
- ArrivalEvent Class, [29](#)
- assign
  - Spinnaker::GenICam::gcstring, [655](#)
- attach
  - Spinnaker::GenApi::FileProtocolAdapter, [643](#)
- AttachBuffer
  - Spinnaker::GenApi::CChunkAdapter, [530](#)
  - Spinnaker::GenApi::CChunkAdapterDcam, [533](#)
  - Spinnaker::GenApi::CChunkAdapterGEV, [536](#)
  - Spinnaker::GenApi::CChunkAdapterGeneric, [534](#), [535](#)
  - Spinnaker::GenApi::CChunkAdapterU3V, [538](#)
- AttachChunk
  - Spinnaker::GenApi::CChunkPort, [540](#)
- AttachEvent
  - Spinnaker::GenApi::CEventPort, [556](#)
- AttachNode
  - Spinnaker::GenApi::CEventPort, [556](#)
- AttachNodeMap

- Spinnaker::GenApi::CChunkAdapter, [530](#)
- Spinnaker::GenApi::CEventAdapter, [547](#)
- AttachPort
  - Spinnaker::GenApi::CChunkPort, [540](#)
- AttachStatistics\_t, [377](#)
  - NumAttachedChunks, [378](#)
  - NumChunkPorts, [378](#)
  - NumChunks, [378](#)
- AutoAlgorithmSelector
  - Spinnaker::Camera, [420](#)
- AutoAlgorithmSelector\_Ae
  - CameraDefs Class, [69](#)
- AutoAlgorithmSelector\_Awb
  - CameraDefs Class, [69](#)
- AutoAlgorithmSelectorEnums
  - CameraDefs Class, [69](#)
- AutoExposureControlLoopDamping
  - Spinnaker::Camera, [420](#)
- AutoExposureControlPriority
  - Spinnaker::Camera, [421](#)
- AutoExposureControlPriority\_ExposureTime
  - CameraDefs Class, [70](#)
- AutoExposureControlPriority\_Gain
  - CameraDefs Class, [70](#)
- AutoExposureControlPriorityEnums
  - CameraDefs Class, [69](#)
- AutoExposureEVCompensation
  - Spinnaker::Camera, [421](#)
- AutoExposureExposureTimeLowerLimit
  - Spinnaker::Camera, [421](#)
- AutoExposureExposureTimeUpperLimit
  - Spinnaker::Camera, [421](#)
- AutoExposureGainLowerLimit
  - Spinnaker::Camera, [421](#)
- AutoExposureGainUpperLimit
  - Spinnaker::Camera, [422](#)
- AutoExposureGreyValueLowerLimit
  - Spinnaker::Camera, [422](#)
- AutoExposureGreyValueUpperLimit
  - Spinnaker::Camera, [422](#)
- AutoExposureLightingMode
  - Spinnaker::Camera, [422](#)
- AutoExposureLightingMode\_AutoDetect
  - CameraDefs Class, [70](#)
- AutoExposureLightingMode\_Backlight
  - CameraDefs Class, [70](#)
- AutoExposureLightingMode\_Frontlight
  - CameraDefs Class, [70](#)
- AutoExposureLightingMode\_Normal
  - CameraDefs Class, [70](#)
- AutoExposureLightingModeEnums
  - CameraDefs Class, [70](#)
- AutoExposureMeteringMode
  - Spinnaker::Camera, [422](#)
- AutoExposureMeteringMode\_Average
  - CameraDefs Class, [70](#)
- AutoExposureMeteringMode\_CenterWeighted
  - CameraDefs Class, [70](#)
- AutoExposureMeteringMode\_HistogramPeak
  - CameraDefs Class, [70](#)
- AutoExposureMeteringMode\_Partial
  - CameraDefs Class, [70](#)
- AutoExposureMeteringMode\_Spot
  - CameraDefs Class, [70](#)
- AutoExposureMeteringModeEnums
  - CameraDefs Class, [70](#)
- AutoExposureTargetGreyValue
  - Spinnaker::Camera, [423](#)
- AutoExposureTargetGreyValueAuto
  - Spinnaker::Camera, [423](#)
- AutoExposureTargetGreyValueAuto\_Continuous
  - CameraDefs Class, [71](#)
- AutoExposureTargetGreyValueAuto\_Off
  - CameraDefs Class, [71](#)
- AutoExposureTargetGreyValueAutoEnums
  - CameraDefs Class, [70](#)
- AutoForcelP
  - Spinnaker::TransportLayerInterface, [834](#)
- AutoLock, [378](#), [379](#)
  - Spinnaker::GenApi::AutoLock, [378](#)
  - Spinnaker::GenICam::AutoLock, [379](#)
- AutoVector Class, [199](#)
- Automatic
  - Types Enums, [314](#)
- BLUE
  - Spinnaker Definitions, [167](#)
- BMPOption, [382](#)
  - Spinnaker::BMPOption, [383](#)
- BMP
  - Spinnaker Definitions, [163](#)
- BalanceRatio
  - Spinnaker::Camera, [423](#)
- BalanceRatioSelector
  - Spinnaker::Camera, [423](#)
- BalanceRatioSelector\_Blue
  - CameraDefs Class, [71](#)
- BalanceRatioSelector\_Red
  - CameraDefs Class, [71](#)
- BalanceRatioSelectorEnums
  - CameraDefs Class, [71](#)
- BalanceWhiteAuto
  - Spinnaker::Camera, [424](#)
- BalanceWhiteAuto\_Continuous
  - CameraDefs Class, [71](#)
- BalanceWhiteAuto\_Off
  - CameraDefs Class, [71](#)
- BalanceWhiteAuto\_Once
  - CameraDefs Class, [71](#)
- BalanceWhiteAutoDamping
  - Spinnaker::Camera, [424](#)
- BalanceWhiteAutoEnums
  - CameraDefs Class, [71](#)
- BalanceWhiteAutoLowerLimit
  - Spinnaker::Camera, [424](#)
- BalanceWhiteAutoProfile
  - Spinnaker::Camera, [424](#)

- BalanceWhiteAutoProfile\_Indoor
  - CameraDefs Class, [71](#)
- BalanceWhiteAutoProfile\_Outdoor
  - CameraDefs Class, [71](#)
- BalanceWhiteAutoProfileEnums
  - CameraDefs Class, [71](#)
- BalanceWhiteAutoUpperLimit
  - Spinnaker::Camera, [424](#)
- BasePtr
  - Spinnaker::BasePtr, [381](#)
- BasePtr Class, [35](#)
- BasePtr< T, B >, [380](#)
- BeginAcquisition
  - Spinnaker::CameraBase, [515](#)
  - Spinnaker::ICameraBase, [669](#)
- Beginner
  - Types Enums, [314](#)
- BigEndian
  - Types Enums, [311](#)
- binaryFile
  - Spinnaker::PGMOption, [789](#)
  - Spinnaker::PPMOption, [799](#)
- BinningHorizontal
  - Spinnaker::Camera, [425](#)
- BinningHorizontalMode
  - Spinnaker::Camera, [425](#)
- BinningHorizontalMode\_Average
  - CameraDefs Class, [72](#)
- BinningHorizontalMode\_Sum
  - CameraDefs Class, [72](#)
- BinningHorizontalModeEnums
  - CameraDefs Class, [71](#)
- BinningSelector
  - Spinnaker::Camera, [425](#)
- BinningSelector\_All
  - CameraDefs Class, [72](#)
- BinningSelector\_ISP
  - CameraDefs Class, [72](#)
- BinningSelector\_Sensor
  - CameraDefs Class, [72](#)
- BinningSelectorEnums
  - CameraDefs Class, [72](#)
- BinningVertical
  - Spinnaker::Camera, [425](#)
- BinningVerticalMode
  - Spinnaker::Camera, [425](#)
- BinningVerticalMode\_Average
  - CameraDefs Class, [72](#)
- BinningVerticalMode\_Sum
  - CameraDefs Class, [72](#)
- BinningVerticalModeEnums
  - CameraDefs Class, [72](#)
- bitrate
  - Spinnaker::Video::H264Option, [665](#)
- BlackLevel
  - Spinnaker::Camera, [425](#)
- BlackLevelAuto
  - Spinnaker::Camera, [426](#)
- BlackLevelAuto\_Continuous
  - CameraDefs Class, [73](#)
- BlackLevelAuto\_Off
  - CameraDefs Class, [73](#)
- BlackLevelAuto\_Once
  - CameraDefs Class, [73](#)
- BlackLevelAutoBalance
  - Spinnaker::Camera, [426](#)
- BlackLevelAutoBalance\_Continuous
  - CameraDefs Class, [72](#)
- BlackLevelAutoBalance\_Off
  - CameraDefs Class, [72](#)
- BlackLevelAutoBalance\_Once
  - CameraDefs Class, [72](#)
- BlackLevelAutoBalanceEnums
  - CameraDefs Class, [72](#)
- BlackLevelAutoEnums
  - CameraDefs Class, [72](#)
- BlackLevelClampingEnable
  - Spinnaker::Camera, [426](#)
- BlackLevelRaw
  - Spinnaker::Camera, [426](#)
- BlackLevelSelector
  - Spinnaker::Camera, [426](#)
- BlackLevelSelector\_All
  - CameraDefs Class, [73](#)
- BlackLevelSelector\_Analog
  - CameraDefs Class, [73](#)
- BlackLevelSelector\_Digital
  - CameraDefs Class, [73](#)
- BlackLevelSelectorEnums
  - CameraDefs Class, [73](#)
- BlockId
  - GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [660](#)
  - GVCP\_EVENT\_ITEM, [658](#)
- BlockId64High
  - GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [660](#)
- BlockId64Low
  - GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [660](#)
- Boolean
  - Types Enums, [313](#)
- BooleanNode, [383](#)
  - Spinnaker::GenApi::BooleanNode, [385](#)
- BooleanNode Class, [203](#)
  - CBooleanRef, [203](#)
- build
  - Spinnaker::LibraryVersion, [758](#)
- c\_str
  - Spinnaker::GenICam::gcstring, [655](#)
- CBasePtr
  - Pointer Class, [289](#)
- CBooleanPtr
  - Pointer Class, [289](#)
- CBooleanRef
  - BooleanNode Class, [203](#)
- CCITTFAX3
  - Spinnaker::TIFFOption, [824](#)
- CCITTFAX4



- Spinnaker::TIFFOption, [824](#)
- CCategoryPtr
  - Pointer Class, [289](#)
- CCategoryRef
  - CategoryNode Class, [204](#)
- CChunkAdapter, [529](#)
  - Spinnaker::GenApi::CChunkAdapter, [530](#)
- CChunkAdapterDcam, [531](#)
  - Spinnaker::GenApi::CChunkAdapterDcam, [532](#)
- CChunkAdapterGEV, [535](#)
  - Spinnaker::GenApi::CChunkAdapterGEV, [536](#)
- CChunkAdapterGeneric, [533](#)
  - Spinnaker::GenApi::CChunkAdapterGeneric, [534](#)
- CChunkAdapterU3V, [537](#)
  - Spinnaker::GenApi::CChunkAdapterU3V, [538](#)
- CChunkPort, [538](#)
  - Spinnaker::GenApi::CChunkPort, [540](#)
- CChunkPortPtr
  - Pointer Class, [290](#)
- CCommandPtr
  - Pointer Class, [290](#)
- CCommandRef
  - CommandNode Class, [210](#)
- CDeviceInfoPtr
  - Pointer Class, [290](#)
- CEnumEntryPtr
  - Pointer Class, [290](#)
- CEnumEntryRef
  - EnumEntryNode Class, [215](#)
- CEnumerationPtr
  - Pointer Class, [290](#)
- CEnumerationRef
  - EnumNode Class, [216](#)
- CEnumerationTRef
  - Spinnaker::GenApi::CEnumerationTRef, [544](#)
- CEnumerationTRef< EnumT >, [542](#)
- CEventAdapter, [546](#)
  - Spinnaker::GenApi::CEventAdapter, [547](#)
- CEventAdapter1394, [548](#)
  - Spinnaker::GenApi::CEventAdapter1394, [549](#)
- CEventAdapterGEV, [551](#)
  - Spinnaker::GenApi::CEventAdapterGEV, [552](#)
- CEventAdapterGeneric, [549](#)
  - Spinnaker::GenApi::CEventAdapterGeneric, [550](#)
- CEventAdapterU3V, [553](#)
  - Spinnaker::GenApi::CEventAdapterU3V, [554](#)
- CEventPort, [554](#)
  - Spinnaker::GenApi::CEventPort, [556](#)
- CFeatureBag, [558](#)
  - Spinnaker::GenApi::CFeatureBag, [559](#)
- CFloatPtr, [560](#)
  - Spinnaker::GenApi::CFloatPtr, [561](#)
- CFloatRef
  - FloatNode Class, [225](#)
- CGeneric\_XMLLoaderParams, [562](#)
- CGlobalLock, [562](#)
  - Spinnaker::GenICam::CGlobalLock, [563](#)
- CGlobalLockUnlocker, [564](#)
  - Spinnaker::GenICam::CGlobalLockUnlocker, [565](#)
- CHUNK\_BASE\_ADDRESS\_REGISTER\_LEN
  - IChunkPort Interface, [238](#)
- CHUNK\_BASE\_ADDRESS\_REGISTER
  - IChunkPort Interface, [238](#)
- CHUNK\_LENGTH\_REGISTER\_LEN
  - IChunkPort Interface, [239](#)
- CHUNK\_LENGTH\_REGISTER
  - IChunkPort Interface, [238](#)
- CIntegerPtr
  - Pointer Class, [290](#)
- CIntegerRef
  - IntegerNode Class, [266](#)
- CLock, [575](#), [577](#)
  - Spinnaker::GenApi::CLock, [576](#)
  - Spinnaker::GenICam::CLock, [577](#)
- CLockEx, [578](#), [579](#)
- CNodeCallback, [580](#)
  - Spinnaker::GenApi::CNodeCallback, [581](#)
- CNodeMapDynPtr
  - Pointer Class, [290](#)
- CNodeMapFactory, [582](#)
  - Spinnaker::GenApi::CNodeMapFactory, [584](#), [585](#)
- CNodeMapFactory::NodeStatistics\_t, [784](#)
- CNodeMapPtr
  - Pointer Class, [290](#)
- CNodeMapRef, [589](#)
  - Spinnaker GenApi Classes, [196](#)
  - Spinnaker::GenApi::CNodeMapRef, [590](#), [591](#)
- CNodeMapRefT< TCameraParams >, [591](#)
- CNodeMapRefT
  - Spinnaker GenApi Classes, [197](#)
- CNodePtr
  - Pointer Class, [290](#)
- CNodeRef
  - Spinnaker GenApi Classes, [196](#)
- COMMAND\_MAGIC
  - Spinnaker::GenApi, [372](#)
- CPointer
  - Spinnaker::GenApi::CPointer, [600](#)
- CPointer< T, B >, [599](#)
- CPortConstructPtr
  - Pointer Class, [290](#)
- CPortImpl, [602](#)
  - Spinnaker::GenApi::CPortImpl, [603](#)
- CPortPtr
  - Pointer Class, [291](#)
- CPortRecorderPtr
  - Pointer Class, [291](#)
- CPortRecorderRef
  - PortRecorder Class, [295](#)
- CPortRef
  - PortNode Class, [294](#)
- CPortReplayPtr
  - Pointer Class, [291](#)
- CPortWriteList, [605](#)
  - Spinnaker::GenApi::CPortWriteList, [606](#)
- CPortWriteListPtr

- Pointer Class, [291](#)
- CRCChecksum
  - DCAM\_CHECKSUM, [615](#)
- CRegisterPortImpl, [607](#)
  - Spinnaker::GenApi::CRegisterPortImpl, [608](#)
- CRegisterPtr
  - Pointer Class, [291](#)
- CRegisterRef
  - RegisterNode Class, [299](#)
- CSelectorPtr
  - Pointer Class, [291](#)
- CSelectorRef
  - Spinnaker GenApi Classes, [196](#)
- CSelectorSet, [609](#)
  - Spinnaker::GenApi::CSelectorSet, [610](#)
- CStringPtr
  - Pointer Class, [291](#)
- CStringRef
  - StringNode Class, [303](#)
- CTestPortStruct
  - Spinnaker::GenApi::CTestPortStruct, [613](#)
- CTestPortStruct< CDataStruct >, [611](#)
- CValuePtr
  - Pointer Class, [291](#)
- CValueRef
  - ValueNode Class, [316](#)
- CacheChunkData
  - IChunkPort Interface, [239](#)
  - Spinnaker::GenApi::PortNode, [792](#)
- CacheUsage\_Automatic
  - NodeMapFactory Class, [284](#)
- CacheUsage\_ForceRead
  - NodeMapFactory Class, [284](#)
- CacheUsage\_ForceWrite
  - NodeMapFactory Class, [284](#)
- CacheUsage\_Ignore
  - NodeMapFactory Class, [284](#)
- CalculateStatistics
  - Spinnaker::Image, [687](#)
  - Spinnaker::Image, [708](#)
- CallbackHandleType
  - Spinnaker GenApi Interfaces, [201](#)
- Camera, [386](#)
  - Spinnaker::Camera, [416](#)
- Camera Base Class, [37](#)
- Camera Base Interface Class, [183](#)
- Camera Class, [36](#)
- Camera List Class, [136](#)
- CameraBase, [513](#)
  - Spinnaker::CameraBase, [515](#)
  - Spinnaker::TransportLayerDevice, [827](#)
  - Spinnaker::TransportLayerStream, [841](#)
- CameraDefs Class, [38](#)
  - AcquisitionMode\_Continuous, [68](#)
  - AcquisitionMode\_MultiFrame, [68](#)
  - AcquisitionMode\_SingleFrame, [68](#)
  - AcquisitionModeEnums, [68](#)
  - AcquisitionStatusSelector\_AcquisitionActive, [69](#)
  - AcquisitionStatusSelector\_AcquisitionTransfer, [69](#)
  - AcquisitionStatusSelector\_AcquisitionTriggerWait, [69](#)
  - AcquisitionStatusSelector\_ExposureActive, [69](#)
  - AcquisitionStatusSelector\_FrameActive, [69](#)
  - AcquisitionStatusSelector\_FrameTriggerWait, [69](#)
  - AcquisitionStatusSelectorEnums, [68](#)
  - ActionUnconditionalMode\_Off, [69](#)
  - ActionUnconditionalMode\_On, [69](#)
  - ActionUnconditionalModeEnums, [69](#)
  - AdcBitDepth\_Bit10, [69](#)
  - AdcBitDepth\_Bit12, [69](#)
  - AdcBitDepth\_Bit14, [69](#)
  - AdcBitDepth\_Bit8, [69](#)
  - AdcBitDepthEnums, [69](#)
  - AutoAlgorithmSelector\_Ae, [69](#)
  - AutoAlgorithmSelector\_Awb, [69](#)
  - AutoAlgorithmSelectorEnums, [69](#)
  - AutoExposureControlPriority\_ExposureTime, [70](#)
  - AutoExposureControlPriority\_Gain, [70](#)
  - AutoExposureControlPriorityEnums, [69](#)
  - AutoExposureLightingMode\_AutoDetect, [70](#)
  - AutoExposureLightingMode\_Backlight, [70](#)
  - AutoExposureLightingMode\_Frontlight, [70](#)
  - AutoExposureLightingMode\_Normal, [70](#)
  - AutoExposureLightingModeEnums, [70](#)
  - AutoExposureMeteringMode\_Average, [70](#)
  - AutoExposureMeteringMode\_CenterWeighted, [70](#)
  - AutoExposureMeteringMode\_HistogramPeak, [70](#)
  - AutoExposureMeteringMode\_Partial, [70](#)
  - AutoExposureMeteringMode\_Spot, [70](#)
  - AutoExposureMeteringModeEnums, [70](#)
  - AutoExposureTargetGreyValueAuto\_Continuous, [71](#)
  - AutoExposureTargetGreyValueAuto\_Off, [71](#)
  - AutoExposureTargetGreyValueAutoEnums, [70](#)
  - BalanceRatioSelector\_Blue, [71](#)
  - BalanceRatioSelector\_Red, [71](#)
  - BalanceRatioSelectorEnums, [71](#)
  - BalanceWhiteAuto\_Continuous, [71](#)
  - BalanceWhiteAuto\_Off, [71](#)
  - BalanceWhiteAuto\_Once, [71](#)
  - BalanceWhiteAutoEnums, [71](#)
  - BalanceWhiteAutoProfile\_Indoor, [71](#)
  - BalanceWhiteAutoProfile\_Outdoor, [71](#)
  - BalanceWhiteAutoProfileEnums, [71](#)
  - BinningHorizontalMode\_Average, [72](#)
  - BinningHorizontalMode\_Sum, [72](#)
  - BinningHorizontalModeEnums, [71](#)
  - BinningSelector\_All, [72](#)
  - BinningSelector\_ISP, [72](#)
  - BinningSelector\_Sensor, [72](#)
  - BinningSelectorEnums, [72](#)
  - BinningVerticalMode\_Average, [72](#)
  - BinningVerticalMode\_Sum, [72](#)
  - BinningVerticalModeEnums, [72](#)
  - BlackLevelAuto\_Continuous, [73](#)
  - BlackLevelAuto\_Off, [73](#)

- BlackLevelAuto\_Once, [73](#)
- BlackLevelAutoBalance\_Continuous, [72](#)
- BlackLevelAutoBalance\_Off, [72](#)
- BlackLevelAutoBalance\_Once, [72](#)
- BlackLevelAutoBalanceEnums, [72](#)
- BlackLevelAutoEnums, [72](#)
- BlackLevelSelector\_All, [73](#)
- BlackLevelSelector\_Analog, [73](#)
- BlackLevelSelector\_Digital, [73](#)
- BlackLevelSelectorEnums, [73](#)
- ChunkBlackLevelSelector\_All, [73](#)
- ChunkBlackLevelSelectorEnums, [73](#)
- ChunkCounterSelector\_Counter0, [73](#)
- ChunkCounterSelector\_Counter1, [73](#)
- ChunkCounterSelector\_Counter2, [73](#)
- ChunkCounterSelectorEnums, [73](#)
- ChunkEncoderSelector\_Encoder0, [74](#)
- ChunkEncoderSelector\_Encoder1, [74](#)
- ChunkEncoderSelector\_Encoder2, [74](#)
- ChunkEncoderSelectorEnums, [73](#)
- ChunkEncoderStatus\_EncoderDown, [74](#)
- ChunkEncoderStatus\_EncoderIdle, [74](#)
- ChunkEncoderStatus\_EncoderStatic, [74](#)
- ChunkEncoderStatus\_EncoderUp, [74](#)
- ChunkEncoderStatusEnums, [74](#)
- ChunkExposureTimeSelector\_Blue, [74](#)
- ChunkExposureTimeSelector\_Common, [74](#)
- ChunkExposureTimeSelector\_Cyan, [74](#)
- ChunkExposureTimeSelector\_Green, [74](#)
- ChunkExposureTimeSelector\_Infrared, [74](#)
- ChunkExposureTimeSelector\_Magenta, [74](#)
- ChunkExposureTimeSelector\_Red, [74](#)
- ChunkExposureTimeSelector\_Stage1, [74](#)
- ChunkExposureTimeSelector\_Stage2, [74](#)
- ChunkExposureTimeSelector\_Ultraviolet, [74](#)
- ChunkExposureTimeSelector\_Yellow, [74](#)
- ChunkExposureTimeSelectorEnums, [74](#)
- ChunkGainSelector\_All, [75](#)
- ChunkGainSelector\_Blue, [75](#)
- ChunkGainSelector\_Green, [75](#)
- ChunkGainSelector\_Red, [75](#)
- ChunkGainSelectorEnums, [74](#)
- ChunkImageComponent\_Color, [75](#)
- ChunkImageComponent\_Confidence, [75](#)
- ChunkImageComponent\_Disparity, [75](#)
- ChunkImageComponent\_Infrared, [75](#)
- ChunkImageComponent\_Intensity, [75](#)
- ChunkImageComponent\_Range, [75](#)
- ChunkImageComponent\_Scatter, [75](#)
- ChunkImageComponent\_Ultraviolet, [75](#)
- ChunkImageComponentEnums, [75](#)
- ChunkPixelFormat\_BayerBG8, [75](#)
- ChunkPixelFormat\_BayerGB8, [75](#)
- ChunkPixelFormat\_BayerGR8, [75](#)
- ChunkPixelFormat\_BayerRG8, [75](#)
- ChunkPixelFormat\_Mono12Packed, [75](#)
- ChunkPixelFormat\_Mono16, [75](#)
- ChunkPixelFormat\_Mono8, [75](#)
- ChunkPixelFormat\_RGB8Packed, [75](#)
- ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY, [75](#)
- ChunkPixelFormat\_YUV422Packed, [75](#)
- ChunkPixelFormatEnums, [75](#)
- ChunkRegionID\_Region0, [76](#)
- ChunkRegionID\_Region1, [76](#)
- ChunkRegionID\_Region2, [76](#)
- ChunkRegionIDEnums, [75](#)
- ChunkScan3dCoordinateReferenceSelector\_↔  
RotationX, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_↔  
RotationY, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_↔  
RotationZ, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_↔  
TranslationX, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_↔  
TranslationY, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_↔  
TranslationZ, [76](#)
- ChunkScan3dCoordinateReferenceSelector↔  
Enums, [76](#)
- ChunkScan3dCoordinateSelector\_CoordinateA, [76](#)
- ChunkScan3dCoordinateSelector\_CoordinateB, [76](#)
- ChunkScan3dCoordinateSelector\_CoordinateC, [76](#)
- ChunkScan3dCoordinateSelectorEnums, [76](#)
- ChunkScan3dCoordinateSystem\_Cartesian, [76](#)
- ChunkScan3dCoordinateSystem\_Cylindrical, [76](#)
- ChunkScan3dCoordinateSystem\_Spherical, [76](#)
- ChunkScan3dCoordinateSystemEnums, [76](#)
- ChunkScan3dCoordinateSystemReference\_↔  
Anchor, [77](#)
- ChunkScan3dCoordinateSystemReference\_↔  
Transformed, [77](#)
- ChunkScan3dCoordinateSystemReferenceEnums, [76](#)
- ChunkScan3dCoordinateTransformSelector\_↔  
RotationX, [77](#)
- ChunkScan3dCoordinateTransformSelector\_↔  
RotationY, [77](#)
- ChunkScan3dCoordinateTransformSelector\_↔  
RotationZ, [77](#)
- ChunkScan3dCoordinateTransformSelector\_↔  
TranslationX, [77](#)
- ChunkScan3dCoordinateTransformSelector\_↔  
TranslationY, [77](#)
- ChunkScan3dCoordinateTransformSelector\_↔  
TranslationZ, [77](#)
- ChunkScan3dCoordinateTransformSelector↔  
Enums, [77](#)
- ChunkScan3dDistanceUnit\_Inch, [77](#)
- ChunkScan3dDistanceUnit\_Millimeter, [77](#)
- ChunkScan3dDistanceUnitEnums, [77](#)
- ChunkScan3dOutputMode\_CalibratedABC\_Grid,

- 78
- ChunkScan3dOutputMode\_CalibratedABC\_↔  
PointCloud, 78
- ChunkScan3dOutputMode\_CalibratedAC\_↔  
Linescan, 78
- ChunkScan3dOutputMode\_CalibratedAC, 78
- ChunkScan3dOutputMode\_CalibratedC\_Linescan,  
78
- ChunkScan3dOutputMode\_CalibratedC, 78
- ChunkScan3dOutputMode\_DisparityC\_Linescan,  
78
- ChunkScan3dOutputMode\_DisparityC, 78
- ChunkScan3dOutputMode\_RectifiedC\_Linescan,  
78
- ChunkScan3dOutputMode\_RectifiedC, 78
- ChunkScan3dOutputMode\_UncalibratedC, 78
- ChunkScan3dOutputModeEnums, 77
- ChunkSelector\_BlackLevel, 79
- ChunkSelector\_CRC, 78
- ChunkSelector\_ExposureEndLineStatusAll, 79
- ChunkSelector\_ExposureTime, 78
- ChunkSelector\_FrameID, 78
- ChunkSelector\_Gain, 79
- ChunkSelector\_Height, 78
- ChunkSelector\_Image, 78
- ChunkSelector\_OffsetX, 78
- ChunkSelector\_OffsetY, 78
- ChunkSelector\_PixelFormat, 79
- ChunkSelector\_SequencerSetActive, 79
- ChunkSelector\_SerialData, 79
- ChunkSelector\_Timestamp, 79
- ChunkSelector\_Width, 78
- ChunkSelectorEnums, 78
- ChunkSourceID\_Source0, 79
- ChunkSourceID\_Source1, 79
- ChunkSourceID\_Source2, 79
- ChunkSourceIDEnums, 79
- ChunkTimerSelector\_Timer0, 79
- ChunkTimerSelector\_Timer1, 79
- ChunkTimerSelector\_Timer2, 79
- ChunkTimerSelectorEnums, 79
- ChunkTransferStreamID\_Stream0, 79
- ChunkTransferStreamID\_Stream1, 79
- ChunkTransferStreamID\_Stream2, 79
- ChunkTransferStreamID\_Stream3, 79
- ChunkTransferStreamIDEnums, 79
- CIConfiguration\_Base, 80
- CIConfiguration\_DualBase, 80
- CIConfiguration\_EightyBit, 80
- CIConfiguration\_Full, 80
- CIConfiguration\_Medium, 80
- CIConfigurationEnums, 79
- CITimeSlotsCount\_One, 80
- CITimeSlotsCount\_Three, 80
- CITimeSlotsCount\_Two, 80
- CITimeSlotsCountEnums, 80
- ColorTransformationSelector\_RGBtoRGB, 80
- ColorTransformationSelector\_RGBtoYUV, 80
- ColorTransformationSelectorEnums, 80
- ColorTransformationValueSelector\_Gain00, 81
- ColorTransformationValueSelector\_Gain01, 81
- ColorTransformationValueSelector\_Gain02, 81
- ColorTransformationValueSelector\_Gain10, 81
- ColorTransformationValueSelector\_Gain11, 81
- ColorTransformationValueSelector\_Gain12, 81
- ColorTransformationValueSelector\_Gain20, 81
- ColorTransformationValueSelector\_Gain21, 81
- ColorTransformationValueSelector\_Gain22, 81
- ColorTransformationValueSelector\_Offset0, 81
- ColorTransformationValueSelector\_Offset1, 81
- ColorTransformationValueSelector\_Offset2, 81
- ColorTransformationValueSelectorEnums, 80
- CounterEventActivation\_AnyEdge, 81
- CounterEventActivation\_FallingEdge, 81
- CounterEventActivation\_LevelHigh, 81
- CounterEventActivation\_LevelLow, 81
- CounterEventActivation\_RisingEdge, 81
- CounterEventActivationEnums, 81
- CounterEventSource\_Counter0End, 82
- CounterEventSource\_Counter0Start, 82
- CounterEventSource\_Counter1End, 82
- CounterEventSource\_Counter1Start, 82
- CounterEventSource\_ExposureEnd, 82
- CounterEventSource\_ExposureStart, 82
- CounterEventSource\_FrameTriggerWait, 82
- CounterEventSource\_Line0, 81
- CounterEventSource\_Line1, 81
- CounterEventSource\_Line2, 81
- CounterEventSource\_Line3, 81
- CounterEventSource\_LogicBlock0, 82
- CounterEventSource\_LogicBlock1, 82
- CounterEventSource\_MHzTick, 81
- CounterEventSource\_Off, 81
- CounterEventSource\_UserOutput0, 81
- CounterEventSource\_UserOutput1, 81
- CounterEventSource\_UserOutput2, 82
- CounterEventSource\_UserOutput3, 82
- CounterEventSourceEnums, 81
- CounterResetActivation\_AnyEdge, 82
- CounterResetActivation\_FallingEdge, 82
- CounterResetActivation\_LevelHigh, 82
- CounterResetActivation\_LevelLow, 82
- CounterResetActivation\_RisingEdge, 82
- CounterResetActivationEnums, 82
- CounterResetSource\_Counter0End, 82
- CounterResetSource\_Counter0Start, 82
- CounterResetSource\_Counter1End, 82
- CounterResetSource\_Counter1Start, 82
- CounterResetSource\_ExposureEnd, 82
- CounterResetSource\_ExposureStart, 82
- CounterResetSource\_FrameTriggerWait, 82
- CounterResetSource\_Line0, 82
- CounterResetSource\_Line1, 82
- CounterResetSource\_Line2, 82
- CounterResetSource\_Line3, 82
- CounterResetSource\_LogicBlock0, 82

CounterResetSource\_LogicBlock1, 82  
CounterResetSource\_Off, 82  
CounterResetSource\_UserOutput0, 82  
CounterResetSource\_UserOutput1, 82  
CounterResetSource\_UserOutput2, 82  
CounterResetSource\_UserOutput3, 82  
CounterResetSourceEnums, 82  
CounterSelector\_Counter0, 83  
CounterSelector\_Counter1, 83  
CounterSelectorEnums, 82  
CounterStatus\_CounterActive, 83  
CounterStatus\_CounterCompleted, 83  
CounterStatus\_CounterIdle, 83  
CounterStatus\_CounterOverflow, 83  
CounterStatus\_CounterTriggerWait, 83  
CounterStatusEnums, 83  
CounterTriggerActivation\_AnyEdge, 83  
CounterTriggerActivation\_FallingEdge, 83  
CounterTriggerActivation\_LevelHigh, 83  
CounterTriggerActivation\_LevelLow, 83  
CounterTriggerActivation\_RisingEdge, 83  
CounterTriggerActivationEnums, 83  
CounterTriggerSource\_Counter0End, 84  
CounterTriggerSource\_Counter0Start, 84  
CounterTriggerSource\_Counter1End, 84  
CounterTriggerSource\_Counter1Start, 84  
CounterTriggerSource\_ExposureEnd, 84  
CounterTriggerSource\_ExposureStart, 84  
CounterTriggerSource\_FrameTriggerWait, 84  
CounterTriggerSource\_Line0, 84  
CounterTriggerSource\_Line1, 84  
CounterTriggerSource\_Line2, 84  
CounterTriggerSource\_Line3, 84  
CounterTriggerSource\_LogicBlock0, 84  
CounterTriggerSource\_LogicBlock1, 84  
CounterTriggerSource\_Off, 84  
CounterTriggerSource\_UserOutput0, 84  
CounterTriggerSource\_UserOutput1, 84  
CounterTriggerSource\_UserOutput2, 84  
CounterTriggerSource\_UserOutput3, 84  
CounterTriggerSourceEnums, 83  
CxpConnectionTestMode\_Mode1, 84  
CxpConnectionTestMode\_Off, 84  
CxpConnectionTestModeEnums, 84  
CxpLinkConfiguration\_Auto, 84  
CxpLinkConfiguration\_CXP1\_X1, 84  
CxpLinkConfiguration\_CXP1\_X2, 85  
CxpLinkConfiguration\_CXP1\_X3, 85  
CxpLinkConfiguration\_CXP1\_X4, 85  
CxpLinkConfiguration\_CXP1\_X5, 85  
CxpLinkConfiguration\_CXP1\_X6, 85  
CxpLinkConfiguration\_CXP2\_X1, 84  
CxpLinkConfiguration\_CXP2\_X2, 85  
CxpLinkConfiguration\_CXP2\_X3, 85  
CxpLinkConfiguration\_CXP2\_X4, 85  
CxpLinkConfiguration\_CXP2\_X5, 85  
CxpLinkConfiguration\_CXP2\_X6, 85  
CxpLinkConfiguration\_CXP3\_X1, 85  
CxpLinkConfiguration\_CXP3\_X2, 85  
CxpLinkConfiguration\_CXP3\_X3, 85  
CxpLinkConfiguration\_CXP3\_X4, 85  
CxpLinkConfiguration\_CXP3\_X5, 85  
CxpLinkConfiguration\_CXP3\_X6, 85  
CxpLinkConfiguration\_CXP5\_X1, 85  
CxpLinkConfiguration\_CXP5\_X2, 85  
CxpLinkConfiguration\_CXP5\_X3, 85  
CxpLinkConfiguration\_CXP5\_X4, 85  
CxpLinkConfiguration\_CXP5\_X5, 85  
CxpLinkConfiguration\_CXP5\_X6, 85  
CxpLinkConfiguration\_CXP6\_X1, 85  
CxpLinkConfiguration\_CXP6\_X2, 85  
CxpLinkConfiguration\_CXP6\_X3, 85  
CxpLinkConfiguration\_CXP6\_X4, 85  
CxpLinkConfiguration\_CXP6\_X5, 85  
CxpLinkConfiguration\_CXP6\_X6, 85  
CxpLinkConfigurationEnums, 84  
CxpLinkConfigurationPreferred\_CXP1\_X1, 85  
CxpLinkConfigurationPreferred\_CXP1\_X2, 86  
CxpLinkConfigurationPreferred\_CXP1\_X3, 86  
CxpLinkConfigurationPreferred\_CXP1\_X4, 86  
CxpLinkConfigurationPreferred\_CXP1\_X5, 86  
CxpLinkConfigurationPreferred\_CXP1\_X6, 86  
CxpLinkConfigurationPreferred\_CXP2\_X1, 85  
CxpLinkConfigurationPreferred\_CXP2\_X2, 86  
CxpLinkConfigurationPreferred\_CXP2\_X3, 86  
CxpLinkConfigurationPreferred\_CXP2\_X4, 86  
CxpLinkConfigurationPreferred\_CXP2\_X5, 86  
CxpLinkConfigurationPreferred\_CXP2\_X6, 86  
CxpLinkConfigurationPreferred\_CXP3\_X1, 85  
CxpLinkConfigurationPreferred\_CXP3\_X2, 86  
CxpLinkConfigurationPreferred\_CXP3\_X3, 86  
CxpLinkConfigurationPreferred\_CXP3\_X4, 86  
CxpLinkConfigurationPreferred\_CXP3\_X5, 86  
CxpLinkConfigurationPreferred\_CXP3\_X6, 86  
CxpLinkConfigurationPreferred\_CXP5\_X1, 85  
CxpLinkConfigurationPreferred\_CXP5\_X2, 86  
CxpLinkConfigurationPreferred\_CXP5\_X3, 86  
CxpLinkConfigurationPreferred\_CXP5\_X4, 86  
CxpLinkConfigurationPreferred\_CXP5\_X5, 86  
CxpLinkConfigurationPreferred\_CXP5\_X6, 86  
CxpLinkConfigurationPreferred\_CXP6\_X1, 86  
CxpLinkConfigurationPreferred\_CXP6\_X2, 86  
CxpLinkConfigurationPreferred\_CXP6\_X3, 86  
CxpLinkConfigurationPreferred\_CXP6\_X4, 86  
CxpLinkConfigurationPreferred\_CXP6\_X5, 86  
CxpLinkConfigurationPreferred\_CXP6\_X6, 86  
CxpLinkConfigurationPreferredEnums, 85  
CxpLinkConfigurationStatus\_CXP1\_X1, 86  
CxpLinkConfigurationStatus\_CXP1\_X2, 86  
CxpLinkConfigurationStatus\_CXP1\_X3, 87  
CxpLinkConfigurationStatus\_CXP1\_X4, 87  
CxpLinkConfigurationStatus\_CXP1\_X5, 87  
CxpLinkConfigurationStatus\_CXP1\_X6, 87  
CxpLinkConfigurationStatus\_CXP2\_X1, 86  
CxpLinkConfigurationStatus\_CXP2\_X2, 86  
CxpLinkConfigurationStatus\_CXP2\_X3, 87



- CxpLinkConfigurationStatus\_CXP2\_X4, [87](#)
- CxpLinkConfigurationStatus\_CXP2\_X5, [87](#)
- CxpLinkConfigurationStatus\_CXP2\_X6, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X1, [86](#)
- CxpLinkConfigurationStatus\_CXP3\_X2, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X3, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X4, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X5, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X6, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X1, [86](#)
- CxpLinkConfigurationStatus\_CXP5\_X2, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X3, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X4, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X5, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X6, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X1, [86](#)
- CxpLinkConfigurationStatus\_CXP6\_X2, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X3, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X4, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X5, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X6, [87](#)
- CxpLinkConfigurationStatus\_None, [86](#)
- CxpLinkConfigurationStatus\_Pending, [86](#)
- CxpLinkConfigurationStatusEnums, [86](#)
- CxpPoCxpStatus\_Auto, [87](#)
- CxpPoCxpStatus\_Off, [87](#)
- CxpPoCxpStatus\_Tripped, [87](#)
- CxpPoCxpStatusEnums, [87](#)
- DecimationHorizontalMode\_Discard, [87](#)
- DecimationHorizontalModeEnums, [87](#)
- DecimationSelector\_All, [88](#)
- DecimationSelector\_Sensor, [88](#)
- DecimationSelectorEnums, [87](#)
- DecimationVerticalMode\_Discard, [88](#)
- DecimationVerticalModeEnums, [88](#)
- DefectCorrectionMode\_Average, [88](#)
- DefectCorrectionMode\_Highlight, [88](#)
- DefectCorrectionMode\_Zero, [88](#)
- DefectCorrectionModeEnums, [88](#)
- Deinterlacing\_LineDuplication, [88](#)
- Deinterlacing\_Off, [88](#)
- Deinterlacing\_Weave, [88](#)
- DeinterlacingEnums, [88](#)
- DeviceCharacterSet\_ASCII, [89](#)
- DeviceCharacterSet\_UTF8, [89](#)
- DeviceCharacterSetEnums, [88](#)
- DeviceClockSelector\_CameraLink, [89](#)
- DeviceClockSelector\_Sensor, [89](#)
- DeviceClockSelector\_SensorDigitization, [89](#)
- DeviceClockSelectorEnums, [89](#)
- DeviceConnectionStatus\_Active, [89](#)
- DeviceConnectionStatus\_Inactive, [89](#)
- DeviceConnectionStatusEnums, [89](#)
- DeviceIndicatorMode\_Active, [89](#)
- DeviceIndicatorMode\_ErrorStatus, [89](#)
- DeviceIndicatorMode\_Inactive, [89](#)
- DeviceIndicatorModeEnums, [89](#)
- DeviceLinkHeartbeatMode\_Off, [90](#)
- DeviceLinkHeartbeatMode\_On, [90](#)
- DeviceLinkHeartbeatModeEnums, [89](#)
- DeviceLinkThroughputLimitMode\_Off, [90](#)
- DeviceLinkThroughputLimitMode\_On, [90](#)
- DeviceLinkThroughputLimitModeEnums, [90](#)
- DevicePowerSupplySelector\_External, [90](#)
- DevicePowerSupplySelectorEnums, [90](#)
- DeviceRegistersEndianness\_Big, [90](#)
- DeviceRegistersEndianness\_Little, [90](#)
- DeviceRegistersEndiannessEnums, [90](#)
- DeviceScanType\_Areascan, [90](#)
- DeviceScanTypeEnums, [90](#)
- DeviceSerialPortBaudRate\_Baud115200, [91](#)
- DeviceSerialPortBaudRate\_Baud19200, [91](#)
- DeviceSerialPortBaudRate\_Baud230400, [91](#)
- DeviceSerialPortBaudRate\_Baud38400, [91](#)
- DeviceSerialPortBaudRate\_Baud460800, [91](#)
- DeviceSerialPortBaudRate\_Baud57600, [91](#)
- DeviceSerialPortBaudRate\_Baud921600, [91](#)
- DeviceSerialPortBaudRate\_Baud9600, [91](#)
- DeviceSerialPortBaudRateEnums, [90](#)
- DeviceSerialPortSelector\_CameraLink, [91](#)
- DeviceSerialPortSelectorEnums, [91](#)
- DeviceStreamChannelEndianness\_Big, [91](#)
- DeviceStreamChannelEndianness\_Little, [91](#)
- DeviceStreamChannelEndiannessEnums, [91](#)
- DeviceStreamChannelType\_Receiver, [91](#)
- DeviceStreamChannelType\_Transmitter, [91](#)
- DeviceStreamChannelTypeEnums, [91](#)
- DeviceTLType\_CameraLink, [93](#)
- DeviceTLType\_CameraLinkHS, [93](#)
- DeviceTLType\_CoaxPress, [93](#)
- DeviceTLType\_Custom, [93](#)
- DeviceTLType\_GigEVision, [93](#)
- DeviceTLType\_USB3Vision, [93](#)
- DeviceTLTypeEnums, [93](#)
- DeviceTapGeometry\_Geometry\_10X\_1Y, [93](#)
- DeviceTapGeometry\_Geometry\_10X, [93](#)
- DeviceTapGeometry\_Geometry\_1X10, [93](#)
- DeviceTapGeometry\_Geometry\_1X10\_1Y, [93](#)
- DeviceTapGeometry\_Geometry\_1X2, [92](#)
- DeviceTapGeometry\_Geometry\_1X2\_1Y2, [92](#)
- DeviceTapGeometry\_Geometry\_1X2\_1Y, [92](#)
- DeviceTapGeometry\_Geometry\_1X2\_2YE, [92](#)
- DeviceTapGeometry\_Geometry\_1X3, [92](#)
- DeviceTapGeometry\_Geometry\_1X3\_1Y, [92](#)
- DeviceTapGeometry\_Geometry\_1X4, [92](#)
- DeviceTapGeometry\_Geometry\_1X4\_1Y, [92](#)
- DeviceTapGeometry\_Geometry\_1X8, [92](#)
- DeviceTapGeometry\_Geometry\_1X8\_1Y, [92](#)
- DeviceTapGeometry\_Geometry\_1X\_1Y2, [92](#)
- DeviceTapGeometry\_Geometry\_1X\_1Y, [92](#)
- DeviceTapGeometry\_Geometry\_1X\_2YE, [92](#)
- DeviceTapGeometry\_Geometry\_1X, [92](#)
- DeviceTapGeometry\_Geometry\_2X2, [92](#)
- DeviceTapGeometry\_Geometry\_2X2\_1Y, [92](#)
- DeviceTapGeometry\_Geometry\_2X2E\_1Y↔  
Geometry\_2X2M\_1Y, [92](#)

- DeviceTapGeometry\_Geometry\_2X2E\_2YE, 92
- DeviceTapGeometry\_Geometry\_2X2E, 92
- DeviceTapGeometry\_Geometry\_2X2M, 92
- DeviceTapGeometry\_Geometry\_2X\_1Y2Geometry↔\_2XE\_1Y, 92
- DeviceTapGeometry\_Geometry\_2X\_1Y, 92
- DeviceTapGeometry\_Geometry\_2X\_2YE, 92
- DeviceTapGeometry\_Geometry\_2XE\_1Y2, 92
- DeviceTapGeometry\_Geometry\_2XE\_2YE, 92
- DeviceTapGeometry\_Geometry\_2XM\_1Y2, 92
- DeviceTapGeometry\_Geometry\_2XM\_1Y, 92
- DeviceTapGeometry\_Geometry\_2XM\_2YE, 92
- DeviceTapGeometry\_Geometry\_2XE, 92
- DeviceTapGeometry\_Geometry\_2XM, 92
- DeviceTapGeometry\_Geometry\_2X, 92
- DeviceTapGeometry\_Geometry\_3X\_1Y, 92
- DeviceTapGeometry\_Geometry\_3X, 92
- DeviceTapGeometry\_Geometry\_4X2, 92
- DeviceTapGeometry\_Geometry\_4X2\_1Y, 92
- DeviceTapGeometry\_Geometry\_4X2E\_1Y, 93
- DeviceTapGeometry\_Geometry\_4X2E, 93
- DeviceTapGeometry\_Geometry\_4X\_1Y, 92
- DeviceTapGeometry\_Geometry\_4X, 92
- DeviceTapGeometry\_Geometry\_8X\_1Y, 92
- DeviceTapGeometry\_Geometry\_8X, 92
- DeviceTapGeometryEnums, 91
- DeviceTemperatureSelector\_Sensor, 93
- DeviceTemperatureSelectorEnums, 93
- DeviceType\_Peripheral, 93
- DeviceType\_Receiver, 93
- DeviceType\_Transceiver, 93
- DeviceType\_Transmitter, 93
- DeviceTypeEnums, 93
- EncoderMode\_FourPhase, 94
- EncoderMode\_HighResolution, 94
- EncoderModeEnums, 93
- EncoderOutputMode\_DirectionDown, 94
- EncoderOutputMode\_DirectionUp, 94
- EncoderOutputMode\_Motion, 94
- EncoderOutputMode\_Off, 94
- EncoderOutputMode\_PositionDown, 94
- EncoderOutputMode\_PositionUp, 94
- EncoderOutputModeEnums, 94
- EncoderResetActivation\_AnyEdge, 94
- EncoderResetActivation\_FallingEdge, 94
- EncoderResetActivation\_LevelHigh, 94
- EncoderResetActivation\_LevelLow, 94
- EncoderResetActivation\_RisingEdge, 94
- EncoderResetActivationEnums, 94
- EncoderResetSource\_AcquisitionEnd, 95
- EncoderResetSource\_AcquisitionStart, 95
- EncoderResetSource\_AcquisitionTrigger, 95
- EncoderResetSource\_Action0, 95
- EncoderResetSource\_Action1, 95
- EncoderResetSource\_Action2, 95
- EncoderResetSource\_Counter0End, 95
- EncoderResetSource\_Counter0Start, 95
- EncoderResetSource\_Counter1End, 95
- EncoderResetSource\_Counter1Start, 95
- EncoderResetSource\_Counter2End, 95
- EncoderResetSource\_Counter2Start, 95
- EncoderResetSource\_ExposureEnd, 95
- EncoderResetSource\_ExposureStart, 95
- EncoderResetSource\_FrameEnd, 95
- EncoderResetSource\_FrameStart, 95
- EncoderResetSource\_FrameTrigger, 95
- EncoderResetSource\_Line0, 95
- EncoderResetSource\_Line1, 95
- EncoderResetSource\_Line2, 95
- EncoderResetSource\_LinkTrigger0, 95
- EncoderResetSource\_LinkTrigger1, 95
- EncoderResetSource\_LinkTrigger2, 95
- EncoderResetSource\_Off, 95
- EncoderResetSource\_SoftwareSignal0, 95
- EncoderResetSource\_SoftwareSignal1, 95
- EncoderResetSource\_SoftwareSignal2, 95
- EncoderResetSource\_Timer0End, 95
- EncoderResetSource\_Timer0Start, 95
- EncoderResetSource\_Timer1End, 95
- EncoderResetSource\_Timer1Start, 95
- EncoderResetSource\_Timer2End, 95
- EncoderResetSource\_Timer2Start, 95
- EncoderResetSource\_UserOutput0, 95
- EncoderResetSource\_UserOutput1, 95
- EncoderResetSource\_UserOutput2, 95
- EncoderResetSourceEnums, 94
- EncoderSelector\_Encoder0, 96
- EncoderSelector\_Encoder1, 96
- EncoderSelector\_Encoder2, 96
- EncoderSelectorEnums, 95
- EncoderSourceA\_Line0, 96
- EncoderSourceA\_Line1, 96
- EncoderSourceA\_Line2, 96
- EncoderSourceA\_Off, 96
- EncoderSourceAEnums, 96
- EncoderSourceB\_Line0, 96
- EncoderSourceB\_Line1, 96
- EncoderSourceB\_Line2, 96
- EncoderSourceB\_Off, 96
- EncoderSourceBEnums, 96
- EncoderStatus\_EncoderDown, 96
- EncoderStatus\_EncoderIdle, 96
- EncoderStatus\_EncoderStatic, 96
- EncoderStatus\_EncoderUp, 96
- EncoderStatusEnums, 96
- EventNotification\_Off, 97
- EventNotification\_On, 97
- EventNotificationEnums, 96
- EventSelector\_Error, 97
- EventSelector\_ExposureEnd, 97
- EventSelector\_SerialPortReceive, 97
- EventSelectorEnums, 97
- ExposureActiveMode\_AllPixels, 97
- ExposureActiveMode\_AnyPixels, 97
- ExposureActiveMode\_Line1, 97
- ExposureActiveModeEnums, 97

- ExposureAuto\_Continuous, 97
- ExposureAuto\_Off, 97
- ExposureAuto\_Once, 97
- ExposureAutoEnums, 97
- ExposureMode\_Timed, 98
- ExposureMode\_TriggerWidth, 98
- ExposureModeEnums, 97
- ExposureTimeMode\_Common, 98
- ExposureTimeMode\_Individual, 98
- ExposureTimeModeEnums, 98
- ExposureTimeSelector\_Blue, 98
- ExposureTimeSelector\_Common, 98
- ExposureTimeSelector\_Cyan, 98
- ExposureTimeSelector\_Green, 98
- ExposureTimeSelector\_Infrared, 98
- ExposureTimeSelector\_Magenta, 98
- ExposureTimeSelector\_Red, 98
- ExposureTimeSelector\_Stage1, 98
- ExposureTimeSelector\_Stage2, 98
- ExposureTimeSelector\_Ultraviolet, 98
- ExposureTimeSelector\_Yellow, 98
- ExposureTimeSelectorEnums, 98
- FileOpenMode\_Read, 99
- FileOpenMode\_ReadWrite, 99
- FileOpenMode\_Write, 99
- FileOpenModeEnums, 98
- FileOperationSelector\_Close, 99
- FileOperationSelector\_Delete, 99
- FileOperationSelector\_Open, 99
- FileOperationSelector\_Read, 99
- FileOperationSelector\_Write, 99
- FileOperationSelectorEnums, 99
- FileOperationStatus\_Failure, 99
- FileOperationStatus\_Overflow, 99
- FileOperationStatus\_Success, 99
- FileOperationStatusEnums, 99
- FileSelector\_SerialPort0, 99
- FileSelector\_UserFile1, 99
- FileSelector\_UserSet0, 99
- FileSelector\_UserSet1, 99
- FileSelector\_UserSetDefault, 99
- FileSelectorEnums, 99
- GainAuto\_Continuous, 100
- GainAuto\_Off, 100
- GainAuto\_Once, 100
- GainAutoBalance\_Continuous, 100
- GainAutoBalance\_Off, 100
- GainAutoBalance\_Once, 100
- GainAutoBalanceEnums, 99
- GainAutoEnums, 100
- GainSelector\_All, 100
- GainSelectorEnums, 100
- GevCCP\_ControlAccess, 100
- GevCCP\_ExclusiveAccess, 100
- GevCCP\_OpenAccess, 100
- GevCCPEnums, 100
- GevCurrentPhysicalLinkConfiguration\_Dynamic↔  
LAG, 101
- GevCurrentPhysicalLinkConfiguration\_MultiLink,  
101
- GevCurrentPhysicalLinkConfiguration\_SingleLink,  
101
- GevCurrentPhysicalLinkConfiguration\_StaticLAG,  
101
- GevCurrentPhysicalLinkConfigurationEnums, 100
- GevGVCPExtendedStatusCodesSelector\_↔  
Version1\_1, 101
- GevGVCPExtendedStatusCodesSelector\_↔  
Version2\_0, 101
- GevGVCPExtendedStatusCodesSelectorEnums,  
101
- GevGVSPExtendedIDMode\_Off, 101
- GevGVSPExtendedIDMode\_On, 101
- GevGVSPExtendedIDModeEnums, 101
- GevIEEE1588ClockAccuracy\_Unknown, 101
- GevIEEE1588ClockAccuracyEnums, 101
- GevIEEE1588Mode\_Auto, 102
- GevIEEE1588Mode\_SlaveOnly, 102
- GevIEEE1588ModeEnums, 101
- GevIEEE1588Status\_Disabled, 102
- GevIEEE1588Status\_Faulty, 102
- GevIEEE1588Status\_Initializing, 102
- GevIEEE1588Status\_Listening, 102
- GevIEEE1588Status\_Master, 102
- GevIEEE1588Status\_Passive, 102
- GevIEEE1588Status\_PreMaster, 102
- GevIEEE1588Status\_Slave, 102
- GevIEEE1588Status\_Uncalibrated, 102
- GevIEEE1588StatusEnums, 102
- GevIPConfigurationStatus\_DHCP, 102
- GevIPConfigurationStatus\_ForceIP, 102
- GevIPConfigurationStatus\_LLA, 102
- GevIPConfigurationStatus\_None, 102
- GevIPConfigurationStatus\_PersistentIP, 102
- GevIPConfigurationStatusEnums, 102
- GevPhysicalLinkConfiguration\_DynamicLAG, 102
- GevPhysicalLinkConfiguration\_MultiLink, 102
- GevPhysicalLinkConfiguration\_SingleLink, 102
- GevPhysicalLinkConfiguration\_StaticLAG, 102
- GevPhysicalLinkConfigurationEnums, 102
- GevSupportedOptionSelector\_Action, 103
- GevSupportedOptionSelector\_CCPApplication↔  
Socket, 103
- GevSupportedOptionSelector\_Commands↔  
Concatenation, 103
- GevSupportedOptionSelector\_DiscoveryAckDelay,  
103
- GevSupportedOptionSelector\_DiscoveryAck↔  
DelayWritable, 103
- GevSupportedOptionSelector\_Event, 103
- GevSupportedOptionSelector\_EventData, 103
- GevSupportedOptionSelector\_ExtendedStatus↔  
Codes, 103
- GevSupportedOptionSelector\_HeartbeatDisable,  
103



- GevSupportedOptionSelector\_IPConfigurationD↔  
HCP, 103
- GevSupportedOptionSelector\_IPConfigurationL↔  
LA, 103
- GevSupportedOptionSelector\_IPConfiguration↔  
PersistentIP, 103
- GevSupportedOptionSelector\_LinkSpeed, 103
- GevSupportedOptionSelector\_ManifestTable, 103
- GevSupportedOptionSelector\_MessageChannel↔  
SourceSocket, 103
- GevSupportedOptionSelector\_PacketResend, 103
- GevSupportedOptionSelector\_PendingAck, 103
- GevSupportedOptionSelector\_SerialNumber, 103
- GevSupportedOptionSelector\_StreamChannel↔  
SourceSocket, 103
- GevSupportedOptionSelector\_TestData, 103
- GevSupportedOptionSelector\_UserDefinedName,  
103
- GevSupportedOptionSelector\_WriteMem, 103
- GevSupportedOptionSelectorEnums, 102
- ImageComponentSelector\_Color, 103
- ImageComponentSelector\_Confidence, 103
- ImageComponentSelector\_Disparity, 103
- ImageComponentSelector\_Infrared, 103
- ImageComponentSelector\_Intensity, 103
- ImageComponentSelector\_Range, 103
- ImageComponentSelector\_Scatter, 104
- ImageComponentSelector\_Ultraviolet, 103
- ImageComponentSelectorEnums, 103
- ImageCompressionJPEGFormatOption\_Baseline↔  
Optimized, 104
- ImageCompressionJPEGFormatOption\_Baseline↔  
Standard, 104
- ImageCompressionJPEGFormatOption\_Lossless,  
104
- ImageCompressionJPEGFormatOption\_Progressive,  
104
- ImageCompressionJPEGFormatOptionEnums,  
104
- ImageCompressionMode\_Lossless, 104
- ImageCompressionMode\_Off, 104
- ImageCompressionModeEnums, 104
- ImageCompressionRateOption\_FixBitrate, 104
- ImageCompressionRateOption\_FixQuality, 104
- ImageCompressionRateOptionEnums, 104
- LUTSelector\_LUT1, 108
- LUTSelectorEnums, 107
- LineFormat\_LVDS, 105
- LineFormat\_NoConnect, 105
- LineFormat\_OpenDrain, 105
- LineFormat\_OptoCoupled, 105
- LineFormat\_RS422, 105
- LineFormat\_TTL, 105
- LineFormat\_TriState, 105
- LineFormatEnums, 104
- LineInputFilterSelector\_Debounce, 105
- LineInputFilterSelector\_Deg glitch, 105
- LineInputFilterSelectorEnums, 105
- LineMode\_Input, 105
- LineMode\_Output, 105
- LineModeEnums, 105
- LineSelector\_Line0, 105
- LineSelector\_Line1, 105
- LineSelector\_Line2, 105
- LineSelector\_Line3, 105
- LineSelectorEnums, 105
- LineSource\_AllPixel, 106
- LineSource\_AnyPixel, 106
- LineSource\_Counter0Active, 106
- LineSource\_Counter1Active, 106
- LineSource\_ExposureActive, 106
- LineSource\_FrameTriggerWait, 106
- LineSource\_Line0, 106
- LineSource\_Line1, 106
- LineSource\_Line2, 106
- LineSource\_Line3, 106
- LineSource\_LogicBlock0, 106
- LineSource\_LogicBlock1, 106
- LineSource\_Off, 106
- LineSource\_PPSSignal, 106
- LineSource\_SerialPort0, 106
- LineSource\_UserOutput0, 106
- LineSource\_UserOutput1, 106
- LineSource\_UserOutput2, 106
- LineSource\_UserOutput3, 106
- LineSourceEnums, 105
- LogicBlockLUTInputActivation\_AnyEdge, 106
- LogicBlockLUTInputActivation\_FallingEdge, 106
- LogicBlockLUTInputActivation\_LevelHigh, 106
- LogicBlockLUTInputActivation\_LevelLow, 106
- LogicBlockLUTInputActivation\_RisingEdge, 106
- LogicBlockLUTInputActivationEnums, 106
- LogicBlockLUTInputSelector\_Input0, 106
- LogicBlockLUTInputSelector\_Input1, 106
- LogicBlockLUTInputSelector\_Input2, 106
- LogicBlockLUTInputSelector\_Input3, 106
- LogicBlockLUTInputSelectorEnums, 106
- LogicBlockLUTInputSource\_AcquisitionActive, 107
- LogicBlockLUTInputSource\_Counter0End, 107
- LogicBlockLUTInputSource\_Counter0Start, 107
- LogicBlockLUTInputSource\_Counter1End, 107
- LogicBlockLUTInputSource\_Counter1Start, 107
- LogicBlockLUTInputSource\_ExposureEnd, 107
- LogicBlockLUTInputSource\_ExposureStart, 107
- LogicBlockLUTInputSource\_FrameTriggerWait,  
107
- LogicBlockLUTInputSource\_Line0, 107
- LogicBlockLUTInputSource\_Line1, 107
- LogicBlockLUTInputSource\_Line2, 107
- LogicBlockLUTInputSource\_Line3, 107
- LogicBlockLUTInputSource\_LogicBlock0, 107
- LogicBlockLUTInputSource\_LogicBlock1, 107
- LogicBlockLUTInputSource\_UserOutput0, 107
- LogicBlockLUTInputSource\_UserOutput1, 107
- LogicBlockLUTInputSource\_UserOutput2, 107
- LogicBlockLUTInputSource\_UserOutput3, 107

- LogicBlockLUTInputSource\_Zero, 107
- LogicBlockLUTInputSourceEnums, 106
- LogicBlockLUTSelector\_Enable, 107
- LogicBlockLUTSelector\_Value, 107
- LogicBlockLUTSelectorEnums, 107
- LogicBlockSelector\_LogicBlock0, 107
- LogicBlockSelector\_LogicBlock1, 107
- LogicBlockSelectorEnums, 107
- NUM\_ACQUISITIONMODE, 68
- NUM\_ACQUISITIONSTATUSSELECTOR, 69
- NUM\_ACTIONUNCONDITIONALMODE, 69
- NUM\_ADCBITDEPTH, 69
- NUM\_AUTOALGORITHMSELECTOR, 69
- NUM\_AUTOEXPOSURECONTROLPRIORITY, 70
- NUM\_AUTOEXPOSURELIGHTINGMODE, 70
- NUM\_AUTOEXPOSUREMETERINGMODE, 70
- NUM\_AUTOEXPOSURETARGETGREYVALUE↔  
AUTO, 71
- NUM\_BALANCERATIOSELECTOR, 71
- NUM\_BALANCEWHITEAUTOPROFILE, 71
- NUM\_BALANCEWHITEAUTO, 71
- NUM\_BINNINGHORIZONTALMODE, 72
- NUM\_BINNINGSELECTOR, 72
- NUM\_BINNINGVERTICALMODE, 72
- NUM\_BLACKLEVELAUTOBALANCE, 72
- NUM\_BLACKLEVELAUTO, 73
- NUM\_BLACKLEVELSELECTOR, 73
- NUM\_CHUNKBLACKLEVELSELECTOR, 73
- NUM\_CHUNKCOUNTERSELECTOR, 73
- NUM\_CHUNKENCODERSELECTOR, 74
- NUM\_CHUNKENCODERSTATUS, 74
- NUM\_CHUNKEXPOSURETIMESELECTOR, 74
- NUM\_CHUNKGAINSELECTOR, 75
- NUM\_CHUNKIMAGECOMPONENT, 75
- NUM\_CHUNKPIXELFORMAT, 75
- NUM\_CHUNKREGIONID, 76
- NUM\_CHUNKSCAN3DCOORDINATEREFERENCE↔  
NCSELECTOR, 76
- NUM\_CHUNKSCAN3DCOORDINATESELECT↔  
OR, 76
- NUM\_CHUNKSCAN3DCOORDINATESYSTEM↔  
REFERENCE, 77
- NUM\_CHUNKSCAN3DCOORDINATESYSTEM,  
76
- NUM\_CHUNKSCAN3DCOORDINATETRANSF↔  
ORMSELECTOR, 77
- NUM\_CHUNKSCAN3DDISTANCEUNIT, 77
- NUM\_CHUNKSCAN3DOUTPUTMODE, 78
- NUM\_CHUNKSELECTOR, 79
- NUM\_CHUNKSOURCEID, 79
- NUM\_CHUNKTIMERSELECTOR, 79
- NUM\_CHUNKTRANSFERSTREAMID, 79
- NUM\_CLCONFIGURATION, 80
- NUM\_CLTIMESLOTSCOUNT, 80
- NUM\_COLORTRANSFORMATIONSELECTOR,  
80
- NUM\_COLORTRANSFORMATIONVALUESEL↔  
ECTOR, 81
- NUM\_COUNTEREVENTACTIVATION, 81
- NUM\_COUNTEREVENTSOURCE, 82
- NUM\_COUNTERRESETACTIVATION, 82
- NUM\_COUNTERRESETSOURCE, 82
- NUM\_COUNTERSELECTOR, 83
- NUM\_COUNTERSTATUS, 83
- NUM\_COUNTERTRIGGERACTIVATION, 83
- NUM\_COUNTERTRIGGERSOURCE, 84
- NUM\_CXPCONNECTIONTESTMODE, 84
- NUM\_CXPLINKCONFIGURATIONPREFERRED,  
86
- NUM\_CXPLINKCONFIGURATIONSTATUS, 87
- NUM\_CXPLINKCONFIGURATION, 85
- NUM\_CXPPOCXPSTATUS, 87
- NUM\_DECIMATIONHORIZONTALMODE, 87
- NUM\_DECIMATIONSELECTOR, 88
- NUM\_DECIMATIONVERTICALMODE, 88
- NUM\_DEFECTCORRECTIONMODE, 88
- NUM\_DEINTERLACING, 88
- NUM\_DEVICECHARACTERSET, 89
- NUM\_DEVICECLOCKSELECTOR, 89
- NUM\_DEVICECONNECTIONSTATUS, 89
- NUM\_DEVICEINDICATORMODE, 89
- NUM\_DEVICELINKHEARTBEATMODE, 90
- NUM\_DEVICELINKTHROUGHPUTLIMITMODE,  
90
- NUM\_DEVICEPOWERSUPPLYSELECTOR, 90
- NUM\_DEVICEREGISTERSENDIANNES, 90
- NUM\_DEVICESCANTYPE, 90
- NUM\_DEVICESERIALPORTBAUDRATE, 91
- NUM\_DEVICESERIALPORTSELECTOR, 91
- NUM\_DEVICESTREAMCHANNELENDIANNES,  
91
- NUM\_DEVICESTREAMCHANNELTYPE, 91
- NUM\_DEVICETAPGEOMETRY, 93
- NUM\_DEVICETEMPERATURESELECTOR, 93
- NUM\_DEVICETLTYPE, 93
- NUM\_DEVICETYPE, 93
- NUM\_ENCODERMODE, 94
- NUM\_ENCODEROUTPUTMODE, 94
- NUM\_ENCODERRESETACTIVATION, 94
- NUM\_ENCODERRESETSOURCE, 95
- NUM\_ENCODERSELECTOR, 96
- NUM\_ENCODERSOURCEA, 96
- NUM\_ENCODERSOURCEB, 96
- NUM\_ENCODERSTATUS, 96
- NUM\_EVENTNOTIFICATION, 97
- NUM\_EVENTSELECTOR, 97
- NUM\_EXPOSUREACTIVEMODE, 97
- NUM\_EXPOSUREAUTO, 97
- NUM\_EXPOSUREMODE, 98
- NUM\_EXPOSURETIMEMODE, 98
- NUM\_EXPOSURETIMESELECTOR, 98
- NUM\_FILEOPENMODE, 99
- NUM\_FILEOPERATIONSELECTOR, 99
- NUM\_FILEOPERATIONSTATUS, 99
- NUM\_FILESELECTOR, 99
- NUM\_GAINAUTOBALANCE, 100

- NUM\_GAINAUTO, [100](#)
- NUM\_GAINSELECTOR, [100](#)
- NUM\_GEVCCP, [100](#)
- NUM\_GEVCURRENTPHYSICALLINKCONFIGURATION, [101](#)
- NUM\_GEVGVCPEXTENDEDSTATUSCODESELECTOR, [101](#)
- NUM\_GEVGVSPEXTENDEDIDMODE, [101](#)
- NUM\_GEVEEEE1588CLOCKACCURACY, [101](#)
- NUM\_GEVEEEE1588MODE, [102](#)
- NUM\_GEVEEEE1588STATUS, [102](#)
- NUM\_GEVIPCONFIGURATIONSTATUS, [102](#)
- NUM\_GEVPHYSCALLINKCONFIGURATION, [102](#)
- NUM\_GEVSUPPORTEDOPTIONSELECTOR, [103](#)
- NUM\_IMAGECOMPONENTSELECTOR, [104](#)
- NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION, [104](#)
- NUM\_IMAGECOMPRESSIONMODE, [104](#)
- NUM\_IMAGECOMPRESSIONRATEOPTION, [104](#)
- NUM\_LINEFORMAT, [105](#)
- NUM\_LINEINPUTFILTERSELECTOR, [105](#)
- NUM\_LINEMODE, [105](#)
- NUM\_LINESELECTOR, [105](#)
- NUM\_LINESOURCE, [106](#)
- NUM\_LOGICBLOCKLUTINPUTACTIVATION, [106](#)
- NUM\_LOGICBLOCKLUTINPUTSELECTOR, [106](#)
- NUM\_LOGICBLOCKLUTINPUTSOURCE, [107](#)
- NUM\_LOGICBLOCKLUTSELECTOR, [107](#)
- NUM\_LOGICBLOCKSELECTOR, [107](#)
- NUM\_LUTSELECTOR, [108](#)
- NUM\_PIXELCOLORFILTER, [108](#)
- NUM\_PIXELFORMATINFOSELECTOR, [119](#)
- NUM\_PIXELFORMAT, [113](#)
- NUM\_PIXELSIZE, [119](#)
- NUM\_REGIONDESTINATION, [119](#)
- NUM\_REGIONMODE, [120](#)
- NUM\_REGIONSELECTOR, [120](#)
- NUM\_RGBTRANSFORMLIGHTSOURCE, [120](#)
- NUM\_SCAN3DCOORDINATEREFERENCESELECTOR, [121](#)
- NUM\_SCAN3DCOORDINATESELECTOR, [121](#)
- NUM\_SCAN3DCOORDINATESYSTEMREFERENCE, [121](#)
- NUM\_SCAN3DCOORDINATESYSTEM, [121](#)
- NUM\_SCAN3DCOORDINATETRANSFORMSELECTOR, [122](#)
- NUM\_SCAN3DDISTANCEUNIT, [122](#)
- NUM\_SCAN3DOUTPUTMODE, [123](#)
- NUM\_SENSORDIGITIZATIONTAPS, [123](#)
- NUM\_SENSORSHUTTERMODE, [123](#)
- NUM\_SENSORTAPS, [123](#)
- NUM\_SEQUENCERCONFIGURATIONMODE, [124](#)
- NUM\_SEQUENCERCONFIGURATIONVALID, [124](#)
- NUM\_SEQUENCERMODE, [124](#)
- NUM\_SEQUENCERSETVALID, [124](#)
- NUM\_SEQUENCERTRIGGERACTIVATION, [124](#)
- NUM\_SEQUENCERTRIGGERSOURCE, [125](#)
- NUM\_SERIALPORTBAUDRATE, [125](#)
- NUM\_SERIALPORTPARITY, [125](#)
- NUM\_SERIALPORTSELECTOR, [126](#)
- NUM\_SERIALPORTSOURCE, [126](#)
- NUM\_SERIALPORTSTOPBITS, [126](#)
- NUM\_SOFTWARESIGNALSELECTOR, [126](#)
- NUM\_SOURCESELECTOR, [127](#)
- NUM\_TESTPATTERNGENERATORSELECTOR, [127](#)
- NUM\_TESTPATTERN, [127](#)
- NUM\_TIMERSELECTOR, [127](#)
- NUM\_TIMERSTATUS, [128](#)
- NUM\_TIMERTRIGGERACTIVATION, [128](#)
- NUM\_TIMERTRIGGERSOURCE, [129](#)
- NUM\_TRANSFERCOMPONENTSELECTOR, [130](#)
- NUM\_TRANSFERCONTROLMODE, [130](#)
- NUM\_TRANSFEROPERATIONMODE, [130](#)
- NUM\_TRANSFERQUEUEMODE, [130](#)
- NUM\_TRANSFERSELECTOR, [131](#)
- NUM\_TRANSFERSTATUSSELECTOR, [131](#)
- NUM\_TRANSFERTRIGGERACTIVATION, [131](#)
- NUM\_TRANSFERTRIGGERMODE, [132](#)
- NUM\_TRANSFERTRIGGERSELECTOR, [132](#)
- NUM\_TRANSFERTRIGGERSOURCE, [133](#)
- NUM\_TRIGGERACTIVATION, [133](#)
- NUM\_TRIGGERMODE, [133](#)
- NUM\_TRIGGEROVERLAP, [134](#)
- NUM\_TRIGGERSELECTOR, [134](#)
- NUM\_TRIGGERSOURCE, [134](#)
- NUM\_USEROUTPUTSELECTOR, [135](#)
- NUM\_USERSETDEFAULT, [135](#)
- NUM\_USERSETSELECTOR, [135](#)
- NUM\_WHITECLIPSELECTOR, [135](#)
- PixelColorFilter\_BayerBG, [108](#)
- PixelColorFilter\_BayerGB, [108](#)
- PixelColorFilter\_BayerGR, [108](#)
- PixelColorFilter\_BayerRG, [108](#)
- PixelColorFilter\_None, [108](#)
- PixelColorFilterEnums, [108](#)
- PixelFormat\_B10, [110](#)
- PixelFormat\_B12, [110](#)
- PixelFormat\_B12\_Jpeg, [113](#)
- PixelFormat\_B16, [110](#)
- PixelFormat\_B8, [110](#)
- PixelFormat\_BGR10, [110](#)
- PixelFormat\_BGR10p, [110](#)
- PixelFormat\_BGR12, [110](#)
- PixelFormat\_BGR12p, [110](#)
- PixelFormat\_BGR14, [110](#)
- PixelFormat\_BGR16, [110](#)
- PixelFormat\_BGR565p, [110](#)
- PixelFormat\_BGR8, [109](#)
- PixelFormat\_BGRa10, [110](#)
- PixelFormat\_BGRa10p, [110](#)
- PixelFormat\_BGRa12, [110](#)

PixelFormat\_BGRa12p, [110](#)  
PixelFormat\_BGRa14, [110](#)  
PixelFormat\_BGRa16, [110](#)  
PixelFormat\_BGRa8, [109](#)  
PixelFormat\_BayerBG10, [109](#)  
PixelFormat\_BayerBG10Packed, [109](#)  
PixelFormat\_BayerBG10p, [109](#)  
PixelFormat\_BayerBG12, [109](#)  
PixelFormat\_BayerBG12Packed, [108](#)  
PixelFormat\_BayerBG12p, [109](#)  
PixelFormat\_BayerBG16, [108](#)  
PixelFormat\_BayerBG8, [108](#)  
PixelFormat\_BayerGB10, [109](#)  
PixelFormat\_BayerGB10Packed, [109](#)  
PixelFormat\_BayerGB10p, [109](#)  
PixelFormat\_BayerGB12, [109](#)  
PixelFormat\_BayerGB12Packed, [108](#)  
PixelFormat\_BayerGB12p, [109](#)  
PixelFormat\_BayerGB16, [108](#)  
PixelFormat\_BayerGB8, [108](#)  
PixelFormat\_BayerGR10, [109](#)  
PixelFormat\_BayerGR10Packed, [109](#)  
PixelFormat\_BayerGR10p, [109](#)  
PixelFormat\_BayerGR12, [109](#)  
PixelFormat\_BayerGR12Packed, [108](#)  
PixelFormat\_BayerGR12p, [109](#)  
PixelFormat\_BayerGR16, [108](#)  
PixelFormat\_BayerGR8, [108](#)  
PixelFormat\_BayerRG10, [109](#)  
PixelFormat\_BayerRG10Packed, [109](#)  
PixelFormat\_BayerRG10p, [109](#)  
PixelFormat\_BayerRG12, [109](#)  
PixelFormat\_BayerRG12Packed, [108](#)  
PixelFormat\_BayerRG12p, [109](#)  
PixelFormat\_BayerRG16, [108](#)  
PixelFormat\_BayerRG8, [108](#)  
PixelFormat\_BayerRGPolarized10p, [113](#)  
PixelFormat\_BayerRGPolarized12p, [113](#)  
PixelFormat\_BayerRGPolarized16, [113](#)  
PixelFormat\_BayerRGPolarized8, [113](#)  
PixelFormat\_BiColorBGRG10, [111](#)  
PixelFormat\_BiColorBGRG10p, [111](#)  
PixelFormat\_BiColorBGRG12, [111](#)  
PixelFormat\_BiColorBGRG12p, [111](#)  
PixelFormat\_BiColorBGRG8, [111](#)  
PixelFormat\_BiColorRGBG10, [111](#)  
PixelFormat\_BiColorRGBG10p, [111](#)  
PixelFormat\_BiColorRGBG12, [111](#)  
PixelFormat\_BiColorRGBG12p, [111](#)  
PixelFormat\_BiColorRGBG8, [111](#)  
PixelFormat\_Confidence1, [111](#)  
PixelFormat\_Confidence16, [111](#)  
PixelFormat\_Confidence1p, [111](#)  
PixelFormat\_Confidence32f, [111](#)  
PixelFormat\_Confidence8, [111](#)  
PixelFormat\_Coord3D\_A10p, [111](#)  
PixelFormat\_Coord3D\_A12p, [111](#)  
PixelFormat\_Coord3D\_A16, [111](#)  
PixelFormat\_Coord3D\_A32f, [111](#)  
PixelFormat\_Coord3D\_A8, [111](#)  
PixelFormat\_Coord3D\_ABC10p, [110](#)  
PixelFormat\_Coord3D\_ABC10p\_Planar, [110](#)  
PixelFormat\_Coord3D\_ABC12p, [110](#)  
PixelFormat\_Coord3D\_ABC12p\_Planar, [110](#)  
PixelFormat\_Coord3D\_ABC16, [110](#)  
PixelFormat\_Coord3D\_ABC16\_Planar, [110](#)  
PixelFormat\_Coord3D\_ABC32f, [110](#)  
PixelFormat\_Coord3D\_ABC32f\_Planar, [111](#)  
PixelFormat\_Coord3D\_ABC8, [110](#)  
PixelFormat\_Coord3D\_ABC8\_Planar, [110](#)  
PixelFormat\_Coord3D\_AC10p, [111](#)  
PixelFormat\_Coord3D\_AC10p\_Planar, [111](#)  
PixelFormat\_Coord3D\_AC12p, [111](#)  
PixelFormat\_Coord3D\_AC12p\_Planar, [111](#)  
PixelFormat\_Coord3D\_AC16, [111](#)  
PixelFormat\_Coord3D\_AC16\_Planar, [111](#)  
PixelFormat\_Coord3D\_AC32f, [111](#)  
PixelFormat\_Coord3D\_AC32f\_Planar, [111](#)  
PixelFormat\_Coord3D\_AC8, [111](#)  
PixelFormat\_Coord3D\_AC8\_Planar, [111](#)  
PixelFormat\_Coord3D\_B10p, [111](#)  
PixelFormat\_Coord3D\_B12p, [111](#)  
PixelFormat\_Coord3D\_B16, [111](#)  
PixelFormat\_Coord3D\_B32f, [111](#)  
PixelFormat\_Coord3D\_B8, [111](#)  
PixelFormat\_Coord3D\_C10p, [111](#)  
PixelFormat\_Coord3D\_C12p, [111](#)  
PixelFormat\_Coord3D\_C16, [111](#)  
PixelFormat\_Coord3D\_C32f, [111](#)  
PixelFormat\_Coord3D\_C8, [111](#)  
PixelFormat\_G10, [110](#)  
PixelFormat\_G12, [110](#)  
PixelFormat\_G16, [110](#)  
PixelFormat\_G8, [110](#)  
PixelFormat\_GB12\_Jpeg, [113](#)  
PixelFormat\_GR12\_Jpeg, [113](#)  
PixelFormat\_Mono10, [109](#)  
PixelFormat\_Mono10Packed, [109](#)  
PixelFormat\_Mono10p, [109](#)  
PixelFormat\_Mono12, [109](#)  
PixelFormat\_Mono12Packed, [108](#)  
PixelFormat\_Mono12p, [109](#)  
PixelFormat\_Mono14, [109](#)  
PixelFormat\_Mono16, [108](#)  
PixelFormat\_Mono1p, [109](#)  
PixelFormat\_Mono2p, [109](#)  
PixelFormat\_Mono4p, [109](#)  
PixelFormat\_Mono8, [108](#)  
PixelFormat\_Mono8s, [109](#)  
PixelFormat\_Polarized10p, [113](#)  
PixelFormat\_Polarized12p, [113](#)  
PixelFormat\_Polarized16, [113](#)  
PixelFormat\_Polarized8, [113](#)  
PixelFormat\_R10, [110](#)  
PixelFormat\_R12, [110](#)  
PixelFormat\_R12\_Jpeg, [113](#)

PixelFormat\_R16, [110](#)  
PixelFormat\_R8, [110](#)  
PixelFormat\_RGB10, [110](#)  
PixelFormat\_RGB10\_Planar, [110](#)  
PixelFormat\_RGB10p, [110](#)  
PixelFormat\_RGB10p32, [110](#)  
PixelFormat\_RGB12, [110](#)  
PixelFormat\_RGB12\_Planar, [110](#)  
PixelFormat\_RGB12p, [110](#)  
PixelFormat\_RGB14, [110](#)  
PixelFormat\_RGB16, [110](#)  
PixelFormat\_RGB16\_Planar, [110](#)  
PixelFormat\_RGB565p, [110](#)  
PixelFormat\_RGB8, [109](#)  
PixelFormat\_RGB8\_Planar, [109](#)  
PixelFormat\_RGB8Packed, [108](#)  
PixelFormat\_RGBa10, [109](#)  
PixelFormat\_RGBa10p, [109](#)  
PixelFormat\_RGBa12, [109](#)  
PixelFormat\_RGBa12p, [109](#)  
PixelFormat\_RGBa14, [109](#)  
PixelFormat\_RGBa16, [109](#)  
PixelFormat\_RGBa8, [109](#)  
PixelFormat\_Raw16, [113](#)  
PixelFormat\_Raw8, [113](#)  
PixelFormat\_SCF1WBWG10, [111](#)  
PixelFormat\_SCF1WBWG10p, [111](#)  
PixelFormat\_SCF1WBWG12, [111](#)  
PixelFormat\_SCF1WBWG12p, [112](#)  
PixelFormat\_SCF1WBWG14, [112](#)  
PixelFormat\_SCF1WBWG16, [112](#)  
PixelFormat\_SCF1WBWG8, [111](#)  
PixelFormat\_SCF1WGWB10, [112](#)  
PixelFormat\_SCF1WGWB10p, [112](#)  
PixelFormat\_SCF1WGWB12, [112](#)  
PixelFormat\_SCF1WGWB12p, [112](#)  
PixelFormat\_SCF1WGWB14, [112](#)  
PixelFormat\_SCF1WGWB16, [112](#)  
PixelFormat\_SCF1WGWB8, [112](#)  
PixelFormat\_SCF1WGWR10, [112](#)  
PixelFormat\_SCF1WGWR10p, [112](#)  
PixelFormat\_SCF1WGWR12, [112](#)  
PixelFormat\_SCF1WGWR12p, [112](#)  
PixelFormat\_SCF1WGWR14, [112](#)  
PixelFormat\_SCF1WGWR16, [112](#)  
PixelFormat\_SCF1WGWR8, [112](#)  
PixelFormat\_SCF1WRWG10, [112](#)  
PixelFormat\_SCF1WRWG10p, [112](#)  
PixelFormat\_SCF1WRWG12, [112](#)  
PixelFormat\_SCF1WRWG12p, [112](#)  
PixelFormat\_SCF1WRWG14, [112](#)  
PixelFormat\_SCF1WRWG16, [112](#)  
PixelFormat\_SCF1WRWG8, [112](#)  
PixelFormat\_YCbCr10\_CbYCr, [112](#)  
PixelFormat\_YCbCr10p\_CbYCr, [112](#)  
PixelFormat\_YCbCr12\_CbYCr, [112](#)  
PixelFormat\_YCbCr12p\_CbYCr, [112](#)  
PixelFormat\_YCbCr411\_8, [109](#)  
PixelFormat\_YCbCr411\_8\_CbYYCrYY, [112](#)  
PixelFormat\_YCbCr422\_10, [112](#)  
PixelFormat\_YCbCr422\_10\_CbYCrY, [112](#)  
PixelFormat\_YCbCr422\_10p, [112](#)  
PixelFormat\_YCbCr422\_10p\_CbYCrY, [112](#)  
PixelFormat\_YCbCr422\_12, [112](#)  
PixelFormat\_YCbCr422\_12\_CbYCrY, [112](#)  
PixelFormat\_YCbCr422\_12p, [112](#)  
PixelFormat\_YCbCr422\_12p\_CbYCrY, [112](#)  
PixelFormat\_YCbCr422\_8, [109](#)  
PixelFormat\_YCbCr422\_8\_CbYCrY, [112](#)  
PixelFormat\_YCbCr601\_10\_CbYCr, [112](#)  
PixelFormat\_YCbCr601\_10p\_CbYCr, [112](#)  
PixelFormat\_YCbCr601\_12\_CbYCr, [112](#)  
PixelFormat\_YCbCr601\_12p\_CbYCr, [112](#)  
PixelFormat\_YCbCr601\_411\_8\_CbYYCrYY, [112](#)  
PixelFormat\_YCbCr601\_422\_10, [113](#)  
PixelFormat\_YCbCr601\_422\_10\_CbYCrY, [113](#)  
PixelFormat\_YCbCr601\_422\_10p, [113](#)  
PixelFormat\_YCbCr601\_422\_10p\_CbYCrY, [113](#)  
PixelFormat\_YCbCr601\_422\_12, [113](#)  
PixelFormat\_YCbCr601\_422\_12\_CbYCrY, [113](#)  
PixelFormat\_YCbCr601\_422\_12p, [113](#)  
PixelFormat\_YCbCr601\_422\_12p\_CbYCrY, [113](#)  
PixelFormat\_YCbCr601\_422\_8, [113](#)  
PixelFormat\_YCbCr601\_422\_8\_CbYCrY, [113](#)  
PixelFormat\_YCbCr601\_8\_CbYCr, [112](#)  
PixelFormat\_YCbCr709\_10\_CbYCr, [113](#)  
PixelFormat\_YCbCr709\_10p\_CbYCr, [113](#)  
PixelFormat\_YCbCr709\_12\_CbYCr, [113](#)  
PixelFormat\_YCbCr709\_12p\_CbYCr, [113](#)  
PixelFormat\_YCbCr709\_411\_8\_CbYYCrYY, [113](#)  
PixelFormat\_YCbCr709\_422\_10, [113](#)  
PixelFormat\_YCbCr709\_422\_10\_CbYCrY, [113](#)  
PixelFormat\_YCbCr709\_422\_10p, [113](#)  
PixelFormat\_YCbCr709\_422\_10p\_CbYCrY, [113](#)  
PixelFormat\_YCbCr709\_422\_12, [113](#)  
PixelFormat\_YCbCr709\_422\_12\_CbYCrY, [113](#)  
PixelFormat\_YCbCr709\_422\_12p, [113](#)  
PixelFormat\_YCbCr709\_422\_12p\_CbYCrY, [113](#)  
PixelFormat\_YCbCr709\_422\_8, [113](#)  
PixelFormat\_YCbCr709\_422\_8\_CbYCrY, [113](#)  
PixelFormat\_YCbCr709\_8\_CbYCr, [113](#)  
PixelFormat\_YCbCr8, [109](#)  
PixelFormat\_YCbCr8\_CbYCr, [112](#)  
PixelFormat\_YUV411\_8\_UYVYY, [113](#)  
PixelFormat\_YUV411Packed, [108](#)  
PixelFormat\_YUV422\_8, [113](#)  
PixelFormat\_YUV422\_8\_UYVY, [113](#)  
PixelFormat\_YUV422Packed, [108](#)  
PixelFormat\_YUV444Packed, [109](#)  
PixelFormat\_YUV8\_UYV, [113](#)  
PixelFormatEnums, [108](#)  
PixelFormatInfoSelector\_B10, [115](#)  
PixelFormatInfoSelector\_B12, [115](#)  
PixelFormatInfoSelector\_B16, [115](#)  
PixelFormatInfoSelector\_B8, [115](#)  
PixelFormatInfoSelector\_BGR10, [115](#)



- PixelFormatInfoSelector\_BGR10p, 115
- PixelFormatInfoSelector\_BGR12, 115
- PixelFormatInfoSelector\_BGR12p, 115
- PixelFormatInfoSelector\_BGR14, 115
- PixelFormatInfoSelector\_BGR16, 115
- PixelFormatInfoSelector\_BGR565p, 115
- PixelFormatInfoSelector\_BGR8, 115
- PixelFormatInfoSelector\_BGRa10, 115
- PixelFormatInfoSelector\_BGRa10p, 115
- PixelFormatInfoSelector\_BGRa12, 115
- PixelFormatInfoSelector\_BGRa12p, 115
- PixelFormatInfoSelector\_BGRa14, 115
- PixelFormatInfoSelector\_BGRa16, 115
- PixelFormatInfoSelector\_BGRa8, 115
- PixelFormatInfoSelector\_BayerBG10, 114
- PixelFormatInfoSelector\_BayerBG10p, 114
- PixelFormatInfoSelector\_BayerBG12, 114
- PixelFormatInfoSelector\_BayerBG12p, 114
- PixelFormatInfoSelector\_BayerBG16, 114
- PixelFormatInfoSelector\_BayerBG8, 114
- PixelFormatInfoSelector\_BayerGB10, 114
- PixelFormatInfoSelector\_BayerGB10p, 114
- PixelFormatInfoSelector\_BayerGB12, 114
- PixelFormatInfoSelector\_BayerGB12p, 114
- PixelFormatInfoSelector\_BayerGB16, 114
- PixelFormatInfoSelector\_BayerGB8, 114
- PixelFormatInfoSelector\_BayerGR10, 114
- PixelFormatInfoSelector\_BayerGR10p, 114
- PixelFormatInfoSelector\_BayerGR12, 114
- PixelFormatInfoSelector\_BayerGR12p, 114
- PixelFormatInfoSelector\_BayerGR16, 114
- PixelFormatInfoSelector\_BayerGR8, 114
- PixelFormatInfoSelector\_BayerRG10, 114
- PixelFormatInfoSelector\_BayerRG10p, 114
- PixelFormatInfoSelector\_BayerRG12, 114
- PixelFormatInfoSelector\_BayerRG12p, 114
- PixelFormatInfoSelector\_BayerRG16, 114
- PixelFormatInfoSelector\_BayerRG8, 114
- PixelFormatInfoSelector\_BayerRGPolarized10p, 119
- PixelFormatInfoSelector\_BayerRGPolarized12p, 119
- PixelFormatInfoSelector\_BayerRGPolarized16, 119
- PixelFormatInfoSelector\_BayerRGPolarized8, 119
- PixelFormatInfoSelector\_BiColorBGRG10, 116
- PixelFormatInfoSelector\_BiColorBGRG10p, 116
- PixelFormatInfoSelector\_BiColorBGRG12, 116
- PixelFormatInfoSelector\_BiColorBGRG12p, 116
- PixelFormatInfoSelector\_BiColorBGRG8, 116
- PixelFormatInfoSelector\_BiColorRGBG10, 116
- PixelFormatInfoSelector\_BiColorRGBG10p, 116
- PixelFormatInfoSelector\_BiColorRGBG12, 116
- PixelFormatInfoSelector\_BiColorRGBG12p, 117
- PixelFormatInfoSelector\_BiColorRGBG8, 116
- PixelFormatInfoSelector\_Confidence1, 116
- PixelFormatInfoSelector\_Confidence16, 116
- PixelFormatInfoSelector\_Confidence1p, 116
- PixelFormatInfoSelector\_Confidence32f, 116
- PixelFormatInfoSelector\_Confidence8, 116
- PixelFormatInfoSelector\_Coord3D\_A10p, 116
- PixelFormatInfoSelector\_Coord3D\_A12p, 116
- PixelFormatInfoSelector\_Coord3D\_A16, 116
- PixelFormatInfoSelector\_Coord3D\_A32f, 116
- PixelFormatInfoSelector\_Coord3D\_A8, 116
- PixelFormatInfoSelector\_Coord3D\_ABC10p, 115
- PixelFormatInfoSelector\_Coord3D\_ABC10p\_↔ Planar, 115
- PixelFormatInfoSelector\_Coord3D\_ABC12p, 116
- PixelFormatInfoSelector\_Coord3D\_ABC12p\_↔ Planar, 116
- PixelFormatInfoSelector\_Coord3D\_ABC16, 116
- PixelFormatInfoSelector\_Coord3D\_ABC16\_↔ Planar, 116
- PixelFormatInfoSelector\_Coord3D\_ABC32f, 116
- PixelFormatInfoSelector\_Coord3D\_ABC32f\_↔ Planar, 116
- PixelFormatInfoSelector\_Coord3D\_ABC8, 115
- PixelFormatInfoSelector\_Coord3D\_ABC8\_Planar, 115
- PixelFormatInfoSelector\_Coord3D\_AC10p, 116
- PixelFormatInfoSelector\_Coord3D\_AC10p\_Planar, 116
- PixelFormatInfoSelector\_Coord3D\_AC12p, 116
- PixelFormatInfoSelector\_Coord3D\_AC12p\_Planar, 116
- PixelFormatInfoSelector\_Coord3D\_AC16, 116
- PixelFormatInfoSelector\_Coord3D\_AC16\_Planar, 116
- PixelFormatInfoSelector\_Coord3D\_AC32f, 116
- PixelFormatInfoSelector\_Coord3D\_AC32f\_Planar, 116
- PixelFormatInfoSelector\_Coord3D\_AC8, 116
- PixelFormatInfoSelector\_Coord3D\_AC8\_Planar, 116
- PixelFormatInfoSelector\_Coord3D\_B10p, 116
- PixelFormatInfoSelector\_Coord3D\_B12p, 116
- PixelFormatInfoSelector\_Coord3D\_B16, 116
- PixelFormatInfoSelector\_Coord3D\_B32f, 116
- PixelFormatInfoSelector\_Coord3D\_B8, 116
- PixelFormatInfoSelector\_Coord3D\_C10p, 116
- PixelFormatInfoSelector\_Coord3D\_C12p, 116
- PixelFormatInfoSelector\_Coord3D\_C16, 116
- PixelFormatInfoSelector\_Coord3D\_C32f, 116
- PixelFormatInfoSelector\_Coord3D\_C8, 116
- PixelFormatInfoSelector\_G10, 115
- PixelFormatInfoSelector\_G12, 115
- PixelFormatInfoSelector\_G16, 115
- PixelFormatInfoSelector\_G8, 115
- PixelFormatInfoSelector\_Mono10, 114
- PixelFormatInfoSelector\_Mono10p, 114
- PixelFormatInfoSelector\_Mono12, 114
- PixelFormatInfoSelector\_Mono12p, 114
- PixelFormatInfoSelector\_Mono14, 114
- PixelFormatInfoSelector\_Mono16, 114
- PixelFormatInfoSelector\_Mono1p, 114

- PixelFormatInfoSelector\_Mono2p, [114](#)
- PixelFormatInfoSelector\_Mono4p, [114](#)
- PixelFormatInfoSelector\_Mono8, [114](#)
- PixelFormatInfoSelector\_Mono8s, [114](#)
- PixelFormatInfoSelector\_Polarized10p, [119](#)
- PixelFormatInfoSelector\_Polarized12p, [119](#)
- PixelFormatInfoSelector\_Polarized16, [119](#)
- PixelFormatInfoSelector\_Polarized8, [119](#)
- PixelFormatInfoSelector\_R10, [115](#)
- PixelFormatInfoSelector\_R12, [115](#)
- PixelFormatInfoSelector\_R16, [115](#)
- PixelFormatInfoSelector\_R8, [115](#)
- PixelFormatInfoSelector\_RGB10, [115](#)
- PixelFormatInfoSelector\_RGB10\_Planar, [115](#)
- PixelFormatInfoSelector\_RGB10p, [115](#)
- PixelFormatInfoSelector\_RGB10p32, [115](#)
- PixelFormatInfoSelector\_RGB12, [115](#)
- PixelFormatInfoSelector\_RGB12\_Planar, [115](#)
- PixelFormatInfoSelector\_RGB12p, [115](#)
- PixelFormatInfoSelector\_RGB14, [115](#)
- PixelFormatInfoSelector\_RGB16, [115](#)
- PixelFormatInfoSelector\_RGB16\_Planar, [115](#)
- PixelFormatInfoSelector\_RGB565p, [115](#)
- PixelFormatInfoSelector\_RGB8, [115](#)
- PixelFormatInfoSelector\_RGB8\_Planar, [115](#)
- PixelFormatInfoSelector\_RGBa10, [114](#)
- PixelFormatInfoSelector\_RGBa10p, [114](#)
- PixelFormatInfoSelector\_RGBa12, [114](#)
- PixelFormatInfoSelector\_RGBa12p, [114](#)
- PixelFormatInfoSelector\_RGBa14, [114](#)
- PixelFormatInfoSelector\_RGBa16, [115](#)
- PixelFormatInfoSelector\_RGBa8, [114](#)
- PixelFormatInfoSelector\_SCF1WBWG10, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG10p, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG12, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG12p, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG14, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG16, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG8, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB10, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB10p, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB12, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB12p, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB14, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB16, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB8, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR10, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR10p, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR12, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR12p, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR14, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR16, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR8, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG10, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG10p, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG12, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG12p, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG14, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG16, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG8, [117](#)
- PixelFormatInfoSelector\_YCbCr10\_CbYCr, [117](#)
- PixelFormatInfoSelector\_YCbCr10p\_CbYCr, [117](#)
- PixelFormatInfoSelector\_YCbCr12\_CbYCr, [117](#)
- PixelFormatInfoSelector\_YCbCr12p\_CbYCr, [117](#)
- PixelFormatInfoSelector\_YCbCr411\_8, [118](#)
- PixelFormatInfoSelector\_YCbCr411\_8\_CbYYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_10, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_10\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_10p, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_10p\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_12, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_12\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_12p, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_12p\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_8, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_8\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_10\_CbYCr, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_10p\_CbYCr, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_12\_CbYCr, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_12p\_CbYCr, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_411\_8\_CbYYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_10, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_10\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_10p, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_10p\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_12, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_12\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_12p, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_12p\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_8, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_8\_CbYCrY, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_8\_CbYCr, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_10\_CbYCr, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_10p\_CbYCr, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_12\_CbYCr, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_12p\_CbYCr, [118](#)

- 118
- PixelFormatInfoSelector\_YCbCr709\_411\_8\_Cb↔YCrY, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_10, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_10\_↔CbYCrY, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_10p, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_10p\_↔CbYCrY, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_12, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_12\_↔CbYCrY, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_12p, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_12p\_↔CbYCrY, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_8, 118
- PixelFormatInfoSelector\_YCbCr709\_422\_8\_Cb↔YCrY, 118
- PixelFormatInfoSelector\_YCbCr709\_8\_CbYCr, 118
- PixelFormatInfoSelector\_YCbCr8, 117
- PixelFormatInfoSelector\_YCbCr8\_CbYCr, 117
- PixelFormatInfoSelector\_YUV411\_8\_UYYVYY, 119
- PixelFormatInfoSelector\_YUV422\_8, 119
- PixelFormatInfoSelector\_YUV422\_8\_UYVY, 119
- PixelFormatInfoSelector\_YUV8\_UYV, 118
- PixelFormatInfoSelectorEnums, 113
- PixelSize\_Bpp1, 119
- PixelSize\_Bpp10, 119
- PixelSize\_Bpp12, 119
- PixelSize\_Bpp14, 119
- PixelSize\_Bpp16, 119
- PixelSize\_Bpp2, 119
- PixelSize\_Bpp20, 119
- PixelSize\_Bpp24, 119
- PixelSize\_Bpp30, 119
- PixelSize\_Bpp32, 119
- PixelSize\_Bpp36, 119
- PixelSize\_Bpp4, 119
- PixelSize\_Bpp48, 119
- PixelSize\_Bpp64, 119
- PixelSize\_Bpp8, 119
- PixelSize\_Bpp96, 119
- PixelSizeEnums, 119
- RegionDestination\_Stream0, 119
- RegionDestination\_Stream1, 119
- RegionDestination\_Stream2, 119
- RegionDestinationEnums, 119
- RegionMode\_Off, 120
- RegionMode\_On, 120
- RegionModeEnums, 119
- RegionSelector\_All, 120
- RegionSelector\_Region0, 120
- RegionSelector\_Region1, 120
- RegionSelector\_Region2, 120
- RegionSelectorEnums, 120
- RgbTransformLightSource\_Cloudy6500K, 120
- RgbTransformLightSource\_CoolFluorescent4000K, 120
- RgbTransformLightSource\_Custom, 120
- RgbTransformLightSource\_Daylight5000K, 120
- RgbTransformLightSource\_General, 120
- RgbTransformLightSource\_Shade8000K, 120
- RgbTransformLightSource\_Tungsten2800K, 120
- RgbTransformLightSource\_WarmFluorescent3000K, 120
- RgbTransformLightSourceEnums, 120
- Scan3dCoordinateReferenceSelector\_RotationX, 121
- Scan3dCoordinateReferenceSelector\_RotationY, 121
- Scan3dCoordinateReferenceSelector\_RotationZ, 121
- Scan3dCoordinateReferenceSelector\_TranslationX, 121
- Scan3dCoordinateReferenceSelector\_TranslationY, 121
- Scan3dCoordinateReferenceSelector\_TranslationZ, 121
- Scan3dCoordinateReferenceSelectorEnums, 120
- Scan3dCoordinateSelector\_CoordinateA, 121
- Scan3dCoordinateSelector\_CoordinateB, 121
- Scan3dCoordinateSelector\_CoordinateC, 121
- Scan3dCoordinateSelectorEnums, 121
- Scan3dCoordinateSystem\_Cartesian, 121
- Scan3dCoordinateSystem\_Cylindrical, 121
- Scan3dCoordinateSystem\_Spherical, 121
- Scan3dCoordinateSystemEnums, 121
- Scan3dCoordinateSystemReference\_Anchor, 121
- Scan3dCoordinateSystemReference\_Transformed, 121
- Scan3dCoordinateSystemReferenceEnums, 121
- Scan3dCoordinateTransformSelector\_RotationX, 122
- Scan3dCoordinateTransformSelector\_RotationY, 122
- Scan3dCoordinateTransformSelector\_RotationZ, 122
- Scan3dCoordinateTransformSelector\_TranslationX, 122
- Scan3dCoordinateTransformSelector\_TranslationY, 122
- Scan3dCoordinateTransformSelector\_TranslationZ, 122
- Scan3dCoordinateTransformSelectorEnums, 121
- Scan3dDistanceUnit\_Inch, 122
- Scan3dDistanceUnit\_Millimeter, 122
- Scan3dDistanceUnitEnums, 122
- Scan3dOutputMode\_CalibratedABC\_Grid, 122
- Scan3dOutputMode\_CalibratedABC\_PointCloud, 122
- Scan3dOutputMode\_CalibratedAC\_Linescan, 122
- Scan3dOutputMode\_CalibratedAC, 122
- Scan3dOutputMode\_CalibratedC\_Linescan, 122
- Scan3dOutputMode\_CalibratedC, 122



- Scan3dOutputMode\_DisparityC\_Linescan, [123](#)
- Scan3dOutputMode\_DisparityC, [123](#)
- Scan3dOutputMode\_RectifiedC\_Linescan, [122](#)
- Scan3dOutputMode\_RectifiedC, [122](#)
- Scan3dOutputMode\_UncalibratedC, [122](#)
- Scan3dOutputModeEnums, [122](#)
- SensorDigitizationTaps\_Eight, [123](#)
- SensorDigitizationTaps\_Four, [123](#)
- SensorDigitizationTaps\_One, [123](#)
- SensorDigitizationTaps\_Ten, [123](#)
- SensorDigitizationTaps\_Three, [123](#)
- SensorDigitizationTaps\_Two, [123](#)
- SensorDigitizationTapsEnums, [123](#)
- SensorShutterMode\_Global, [123](#)
- SensorShutterMode\_GlobalReset, [123](#)
- SensorShutterMode\_Rolling, [123](#)
- SensorShutterModeEnums, [123](#)
- SensorTaps\_Eight, [123](#)
- SensorTaps\_Four, [123](#)
- SensorTaps\_One, [123](#)
- SensorTaps\_Ten, [123](#)
- SensorTaps\_Three, [123](#)
- SensorTaps\_Two, [123](#)
- SensorTapsEnums, [123](#)
- SequencerConfigurationMode\_Off, [124](#)
- SequencerConfigurationMode\_On, [124](#)
- SequencerConfigurationModeEnums, [123](#)
- SequencerConfigurationValid\_No, [124](#)
- SequencerConfigurationValid\_Yes, [124](#)
- SequencerConfigurationValidEnums, [124](#)
- SequencerMode\_Off, [124](#)
- SequencerMode\_On, [124](#)
- SequencerModeEnums, [124](#)
- SequencerSetValid\_No, [124](#)
- SequencerSetValid\_Yes, [124](#)
- SequencerSetValidEnums, [124](#)
- SequencerTriggerActivation\_AnyEdge, [124](#)
- SequencerTriggerActivation\_FallingEdge, [124](#)
- SequencerTriggerActivation\_LevelHigh, [124](#)
- SequencerTriggerActivation\_LevelLow, [124](#)
- SequencerTriggerActivation\_RisingEdge, [124](#)
- SequencerTriggerActivationEnums, [124](#)
- SequencerTriggerSource\_FrameStart, [125](#)
- SequencerTriggerSource\_Off, [125](#)
- SequencerTriggerSourceEnums, [124](#)
- SerialPortBaudRate\_Baud115200, [125](#)
- SerialPortBaudRate\_Baud1200, [125](#)
- SerialPortBaudRate\_Baud14400, [125](#)
- SerialPortBaudRate\_Baud19200, [125](#)
- SerialPortBaudRate\_Baud230400, [125](#)
- SerialPortBaudRate\_Baud2400, [125](#)
- SerialPortBaudRate\_Baud300, [125](#)
- SerialPortBaudRate\_Baud38400, [125](#)
- SerialPortBaudRate\_Baud460800, [125](#)
- SerialPortBaudRate\_Baud4800, [125](#)
- SerialPortBaudRate\_Baud57600, [125](#)
- SerialPortBaudRate\_Baud600, [125](#)
- SerialPortBaudRate\_Baud921600, [125](#)
- SerialPortBaudRate\_Baud9600, [125](#)
- SerialPortBaudRateEnums, [125](#)
- SerialPortParity\_Even, [125](#)
- SerialPortParity\_Mark, [125](#)
- SerialPortParity\_None, [125](#)
- SerialPortParity\_Odd, [125](#)
- SerialPortParity\_Space, [125](#)
- SerialPortParityEnums, [125](#)
- SerialPortSelector\_SerialPort0, [126](#)
- SerialPortSelectorEnums, [125](#)
- SerialPortSource\_Line0, [126](#)
- SerialPortSource\_Line1, [126](#)
- SerialPortSource\_Line2, [126](#)
- SerialPortSource\_Line3, [126](#)
- SerialPortSource\_Off, [126](#)
- SerialPortSourceEnums, [126](#)
- SerialPortStopBits\_Bits1, [126](#)
- SerialPortStopBits\_Bits1AndAHalf, [126](#)
- SerialPortStopBits\_Bits2, [126](#)
- SerialPortStopBitsEnums, [126](#)
- SoftwareSignalSelector\_SoftwareSignal0, [126](#)
- SoftwareSignalSelector\_SoftwareSignal1, [126](#)
- SoftwareSignalSelector\_SoftwareSignal2, [126](#)
- SoftwareSignalSelectorEnums, [126](#)
- SourceSelector\_All, [127](#)
- SourceSelector\_Source0, [127](#)
- SourceSelector\_Source1, [127](#)
- SourceSelector\_Source2, [127](#)
- SourceSelectorEnums, [126](#)
- TestPattern\_Increment, [127](#)
- TestPattern\_Off, [127](#)
- TestPattern\_SensorTestPattern, [127](#)
- TestPatternEnums, [127](#)
- TestPatternGeneratorSelector\_PipelineStart, [127](#)
- TestPatternGeneratorSelector\_Sensor, [127](#)
- TestPatternGeneratorSelectorEnums, [127](#)
- TimerSelector\_Timer0, [127](#)
- TimerSelector\_Timer1, [127](#)
- TimerSelector\_Timer2, [127](#)
- TimerSelectorEnums, [127](#)
- TimerStatus\_TimerActive, [128](#)
- TimerStatus\_TimerCompleted, [128](#)
- TimerStatus\_TimerIdle, [128](#)
- TimerStatus\_TimerTriggerWait, [128](#)
- TimerStatusEnums, [127](#)
- TimerTriggerActivation\_AnyEdge, [128](#)
- TimerTriggerActivation\_FallingEdge, [128](#)
- TimerTriggerActivation\_LevelHigh, [128](#)
- TimerTriggerActivation\_LevelLow, [128](#)
- TimerTriggerActivation\_RisingEdge, [128](#)
- TimerTriggerActivationEnums, [128](#)
- TimerTriggerSource\_AcquisitionEnd, [128](#)
- TimerTriggerSource\_AcquisitionStart, [128](#)
- TimerTriggerSource\_Action0, [129](#)
- TimerTriggerSource\_Action1, [129](#)
- TimerTriggerSource\_Action2, [129](#)
- TimerTriggerSource\_Counter0End, [129](#)

- TimerTriggerSource\_Counter0Start, 129
- TimerTriggerSource\_Counter1End, 129
- TimerTriggerSource\_Counter1Start, 129
- TimerTriggerSource\_Counter2End, 129
- TimerTriggerSource\_Counter2Start, 129
- TimerTriggerSource\_Encoder0, 129
- TimerTriggerSource\_Encoder1, 129
- TimerTriggerSource\_Encoder2, 129
- TimerTriggerSource\_ExposureEnd, 128
- TimerTriggerSource\_ExposureStart, 128
- TimerTriggerSource\_FrameBurstEnd, 128
- TimerTriggerSource\_FrameBurstStart, 128
- TimerTriggerSource\_FrameEnd, 128
- TimerTriggerSource\_FrameStart, 128
- TimerTriggerSource\_FrameTrigger, 128
- TimerTriggerSource\_Line0, 128
- TimerTriggerSource\_Line1, 129
- TimerTriggerSource\_Line2, 129
- TimerTriggerSource\_LineEnd, 128
- TimerTriggerSource\_LineStart, 128
- TimerTriggerSource\_LineTrigger, 128
- TimerTriggerSource\_LinkTrigger0, 129
- TimerTriggerSource\_LinkTrigger1, 129
- TimerTriggerSource\_LinkTrigger2, 129
- TimerTriggerSource\_Off, 128
- TimerTriggerSource\_SoftwareSignal0, 129
- TimerTriggerSource\_SoftwareSignal1, 129
- TimerTriggerSource\_SoftwareSignal2, 129
- TimerTriggerSource\_Timer0End, 129
- TimerTriggerSource\_Timer0Start, 129
- TimerTriggerSource\_Timer1End, 129
- TimerTriggerSource\_Timer1Start, 129
- TimerTriggerSource\_Timer2End, 129
- TimerTriggerSource\_Timer2Start, 129
- TimerTriggerSource\_UserOutput0, 129
- TimerTriggerSource\_UserOutput1, 129
- TimerTriggerSource\_UserOutput2, 129
- TimerTriggerSourceEnums, 128
- TransferComponentSelector\_All, 130
- TransferComponentSelector\_Blue, 130
- TransferComponentSelector\_Green, 130
- TransferComponentSelector\_Red, 130
- TransferComponentSelectorEnums, 129
- TransferControlMode\_Automatic, 130
- TransferControlMode\_Basic, 130
- TransferControlMode\_UserControlled, 130
- TransferControlModeEnums, 130
- TransferOperationMode\_Continuous, 130
- TransferOperationMode\_MultiBlock, 130
- TransferOperationModeEnums, 130
- TransferQueueMode\_FirstInFirstOut, 130
- TransferQueueModeEnums, 130
- TransferSelector\_All, 131
- TransferSelector\_Stream0, 131
- TransferSelector\_Stream1, 131
- TransferSelector\_Stream2, 131
- TransferSelectorEnums, 130
- TransferStatusSelector\_Paused, 131
- TransferStatusSelector\_QueueOverflow, 131
- TransferStatusSelector\_Stopped, 131
- TransferStatusSelector\_Stopping, 131
- TransferStatusSelector\_Streaming, 131
- TransferStatusSelectorEnums, 131
- TransferTriggerActivation\_AnyEdge, 131
- TransferTriggerActivation\_FallingEdge, 131
- TransferTriggerActivation\_LevelHigh, 131
- TransferTriggerActivation\_LevelLow, 131
- TransferTriggerActivation\_RisingEdge, 131
- TransferTriggerActivationEnums, 131
- TransferTriggerMode\_Off, 132
- TransferTriggerMode\_On, 132
- TransferTriggerModeEnums, 131
- TransferTriggerSelector\_TransferAbort, 132
- TransferTriggerSelector\_TransferActive, 132
- TransferTriggerSelector\_TransferBurstStart, 132
- TransferTriggerSelector\_TransferBurstStop, 132
- TransferTriggerSelector\_TransferPause, 132
- TransferTriggerSelector\_TransferResume, 132
- TransferTriggerSelector\_TransferStart, 132
- TransferTriggerSelector\_TransferStop, 132
- TransferTriggerSelectorEnums, 132
- TransferTriggerSource\_Action0, 133
- TransferTriggerSource\_Action1, 133
- TransferTriggerSource\_Action2, 133
- TransferTriggerSource\_Counter0End, 132
- TransferTriggerSource\_Counter0Start, 132
- TransferTriggerSource\_Counter1End, 132
- TransferTriggerSource\_Counter1Start, 132
- TransferTriggerSource\_Counter2End, 133
- TransferTriggerSource\_Counter2Start, 132
- TransferTriggerSource\_Line0, 132
- TransferTriggerSource\_Line1, 132
- TransferTriggerSource\_Line2, 132
- TransferTriggerSource\_SoftwareSignal0, 133
- TransferTriggerSource\_SoftwareSignal1, 133
- TransferTriggerSource\_SoftwareSignal2, 133
- TransferTriggerSource\_Timer0End, 133
- TransferTriggerSource\_Timer0Start, 133
- TransferTriggerSource\_Timer1End, 133
- TransferTriggerSource\_Timer1Start, 133
- TransferTriggerSource\_Timer2End, 133
- TransferTriggerSource\_Timer2Start, 133
- TransferTriggerSourceEnums, 132
- TriggerActivation\_AnyEdge, 133
- TriggerActivation\_FallingEdge, 133
- TriggerActivation\_LevelHigh, 133
- TriggerActivation\_LevelLow, 133
- TriggerActivation\_RisingEdge, 133
- TriggerActivationEnums, 133
- TriggerMode\_Off, 133
- TriggerMode\_On, 133
- TriggerModeEnums, 133
- TriggerOverlap\_Off, 134
- TriggerOverlap\_PreviousFrame, 134
- TriggerOverlap\_ReadOut, 134
- TriggerOverlapEnums, 133

- TriggerSelector\_AcquisitionStart, [134](#)
- TriggerSelector\_FrameBurstStart, [134](#)
- TriggerSelector\_FrameStart, [134](#)
- TriggerSelectorEnums, [134](#)
- TriggerSource\_Action0, [134](#)
- TriggerSource\_Counter0End, [134](#)
- TriggerSource\_Counter0Start, [134](#)
- TriggerSource\_Counter1End, [134](#)
- TriggerSource\_Counter1Start, [134](#)
- TriggerSource\_Line0, [134](#)
- TriggerSource\_Line1, [134](#)
- TriggerSource\_Line2, [134](#)
- TriggerSource\_Line3, [134](#)
- TriggerSource\_LogicBlock0, [134](#)
- TriggerSource\_LogicBlock1, [134](#)
- TriggerSource\_Software, [134](#)
- TriggerSource\_UserOutput0, [134](#)
- TriggerSource\_UserOutput1, [134](#)
- TriggerSource\_UserOutput2, [134](#)
- TriggerSource\_UserOutput3, [134](#)
- TriggerSourceEnums, [134](#)
- UNKNOWN\_PIXELFORMAT, [113](#)
- UserOutputSelector\_UserOutput0, [135](#)
- UserOutputSelector\_UserOutput1, [135](#)
- UserOutputSelector\_UserOutput2, [135](#)
- UserOutputSelector\_UserOutput3, [135](#)
- UserOutputSelectorEnums, [134](#)
- UserSetDefault\_Default, [135](#)
- UserSetDefault\_UserSet0, [135](#)
- UserSetDefault\_UserSet1, [135](#)
- UserSetDefaultEnums, [135](#)
- UserSetSelector\_Default, [135](#)
- UserSetSelector\_UserSet0, [135](#)
- UserSetSelector\_UserSet1, [135](#)
- UserSetSelectorEnums, [135](#)
- WhiteClipSelector\_All, [135](#)
- WhiteClipSelector\_Blue, [135](#)
- WhiteClipSelector\_Green, [135](#)
- WhiteClipSelector\_Red, [135](#)
- WhiteClipSelector\_Tap1, [135](#)
- WhiteClipSelector\_Tap2, [135](#)
- WhiteClipSelector\_U, [135](#)
- WhiteClipSelector\_V, [135](#)
- WhiteClipSelector\_Y, [135](#)
- WhiteClipSelectorEnums, [135](#)
- CameraInternal
  - Spinnaker::ICameraBase, [672](#)
  - Spinnaker::TransportLayerDevice, [827](#)
  - Spinnaker::TransportLayerStream, [841](#)
- CameraList, [522](#)
  - Spinnaker::CameraList, [523](#)
- CameraListImpl
  - Spinnaker::ICameraList, [674](#)
- CameraPtr, [526](#)
  - CameraPtr Class, [137](#)
- CameraPtr Class, [137](#)
  - ~CameraPtr, [137](#)
  - CameraPtr, [137](#)
  - operator=, [137](#)
- CastToIDestroy
  - Spinnaker GenApi Classes, [197](#)
- CategoryNode, [527](#)
  - Spinnaker::GenApi::CategoryNode, [528](#)
- CategoryNode Class, [204](#)
  - CCategoryRef, [204](#)
- cbPostInsideLock
  - NodeCallback Class, [282](#)
- cbPostOutsideLock
  - NodeCallback Class, [282](#)
- CheckBufferLayout
  - Spinnaker::GenApi::CChunkAdapter, [530](#)
  - Spinnaker::GenApi::CChunkAdapterDcam, [533](#)
  - Spinnaker::GenApi::CChunkAdapterGEV, [536](#)
  - Spinnaker::GenApi::CChunkAdapterGeneric, [535](#)
  - Spinnaker::GenApi::CChunkAdapterU3V, [538](#)
- CheckCRC
  - Spinnaker::GenApi::CChunkAdapterDcam, [533](#)
  - Spinnaker::Image, [687](#)
  - Spinnaker::Image, [708](#)
- CheckChunkID
  - Spinnaker::GenApi::CChunkPort, [540](#)
- CheckEventID
  - Spinnaker::GenApi::CEventPort, [556](#)
- ChunkAdapter Class, [205](#)
- ChunkAdapterDcam Class, [206](#)
- ChunkAdapterGEV Class, [208](#)
- ChunkAdapterGeneric Class, [207](#)
- ChunkAdapterU3V Class, [317](#)
- ChunkBlackLevel
  - Spinnaker::Camera, [426](#)
- ChunkBlackLevelSelector
  - Spinnaker::Camera, [427](#)
- ChunkBlackLevelSelector\_All
  - CameraDefs Class, [73](#)
- ChunkBlackLevelSelectorEnums
  - CameraDefs Class, [73](#)
- ChunkCRC
  - Spinnaker::Camera, [427](#)
- ChunkCounterSelector
  - Spinnaker::Camera, [427](#)
- ChunkCounterSelector\_Counter0
  - CameraDefs Class, [73](#)
- ChunkCounterSelector\_Counter1
  - CameraDefs Class, [73](#)
- ChunkCounterSelector\_Counter2
  - CameraDefs Class, [73](#)
- ChunkCounterSelectorEnums
  - CameraDefs Class, [73](#)
- ChunkCounterValue
  - Spinnaker::Camera, [427](#)
- ChunkData, [566](#)
  - Spinnaker::ChunkData, [568](#)
- ChunkData Class, [138](#)
- ChunkEnable
  - Spinnaker::Camera, [427](#)
- ChunkEncoderSelector

- Spinnaker::Camera, [427](#)
- ChunkEncoderSelector\_Encoder0
  - CameraDefs Class, [74](#)
- ChunkEncoderSelector\_Encoder1
  - CameraDefs Class, [74](#)
- ChunkEncoderSelector\_Encoder2
  - CameraDefs Class, [74](#)
- ChunkEncoderSelectorEnums
  - CameraDefs Class, [73](#)
- ChunkEncoderStatus
  - Spinnaker::Camera, [427](#)
- ChunkEncoderStatus\_EncoderDown
  - CameraDefs Class, [74](#)
- ChunkEncoderStatus\_EncoderIdle
  - CameraDefs Class, [74](#)
- ChunkEncoderStatus\_EncoderStatic
  - CameraDefs Class, [74](#)
- ChunkEncoderStatus\_EncoderUp
  - CameraDefs Class, [74](#)
- ChunkEncoderStatusEnums
  - CameraDefs Class, [74](#)
- ChunkEncoderValue
  - Spinnaker::Camera, [427](#)
- ChunkExposureEndLineStatusAll
  - Spinnaker::Camera, [428](#)
- ChunkExposureTime
  - Spinnaker::Camera, [428](#)
- ChunkExposureTimeSelector
  - Spinnaker::Camera, [428](#)
- ChunkExposureTimeSelector\_Blue
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Common
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Cyan
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Green
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Infrared
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Magenta
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Red
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Stage1
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Stage2
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Ultraviolet
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelector\_Yellow
  - CameraDefs Class, [74](#)
- ChunkExposureTimeSelectorEnums
  - CameraDefs Class, [74](#)
- ChunkFrameID
  - Spinnaker::Camera, [428](#)
- ChunkGain
  - Spinnaker::Camera, [428](#)
- ChunkGainSelector
  - Spinnaker::Camera, [428](#)
- Spinnaker::Camera, [428](#)
- ChunkGainSelector\_All
  - CameraDefs Class, [75](#)
- ChunkGainSelector\_Blue
  - CameraDefs Class, [75](#)
- ChunkGainSelector\_Green
  - CameraDefs Class, [75](#)
- ChunkGainSelector\_Red
  - CameraDefs Class, [75](#)
- ChunkGainSelectorEnums
  - CameraDefs Class, [74](#)
- ChunkHeight
  - Spinnaker::Camera, [428](#)
- ChunkID
  - DCAM\_CHUNK\_TRAILER, [615](#)
  - GVCP\_CHUNK\_TRAILER, [658](#)
  - SingleChunkData\_t, [805](#)
  - SingleChunkDataStr\_t, [805](#)
  - U3V\_CHUNK\_TRAILER, [844](#)
- ChunkImage
  - Spinnaker::Camera, [428](#)
- ChunkImageComponent
  - Spinnaker::Camera, [429](#)
- ChunkImageComponent\_Color
  - CameraDefs Class, [75](#)
- ChunkImageComponent\_Confidence
  - CameraDefs Class, [75](#)
- ChunkImageComponent\_Disparity
  - CameraDefs Class, [75](#)
- ChunkImageComponent\_Infrared
  - CameraDefs Class, [75](#)
- ChunkImageComponent\_Intensity
  - CameraDefs Class, [75](#)
- ChunkImageComponent\_Range
  - CameraDefs Class, [75](#)
- ChunkImageComponent\_Scatter
  - CameraDefs Class, [75](#)
- ChunkImageComponent\_Ultraviolet
  - CameraDefs Class, [75](#)
- ChunkImageComponentEnums
  - CameraDefs Class, [75](#)
- ChunkInferenceConfidence
  - Spinnaker::Camera, [429](#)
- ChunkInferenceResult
  - Spinnaker::Camera, [429](#)
- ChunkLength
  - DCAM\_CHUNK\_TRAILER, [615](#)
  - GVCP\_CHUNK\_TRAILER, [658](#)
  - SingleChunkData\_t, [805](#)
  - SingleChunkDataStr\_t, [805](#)
  - U3V\_CHUNK\_TRAILER, [844](#)
- ChunkLinePitch
  - Spinnaker::Camera, [429](#)
- ChunkLineStatusAll
  - Spinnaker::Camera, [429](#)
- ChunkModeActive
  - Spinnaker::Camera, [429](#)
- ChunkOffset

- SingleChunkData\_t, [805](#)
- SingleChunkDataStr\_t, [805](#)
- ChunkOffsetX
  - Spinnaker::Camera, [429](#)
- ChunkOffsetY
  - Spinnaker::Camera, [429](#)
- ChunkPartSelector
  - Spinnaker::Camera, [430](#)
- ChunkPixelDynamicRangeMax
  - Spinnaker::Camera, [430](#)
- ChunkPixelDynamicRangeMin
  - Spinnaker::Camera, [430](#)
- ChunkPixelFormat
  - Spinnaker::Camera, [430](#)
- ChunkPixelFormat\_BayerBG8
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_BayerGB8
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_BayerGR8
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_BayerRG8
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_Mono12Packed
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_Mono16
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_Mono8
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_RGB8Packed
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY
  - CameraDefs Class, [75](#)
- ChunkPixelFormat\_YUV422Packed
  - CameraDefs Class, [75](#)
- ChunkPixelFormatEnums
  - CameraDefs Class, [75](#)
- ChunkPort Class, [209](#)
- ChunkRegionID\_Region0
  - CameraDefs Class, [76](#)
- ChunkRegionID\_Region1
  - CameraDefs Class, [76](#)
- ChunkRegionID\_Region2
  - CameraDefs Class, [76](#)
- ChunkRegionIDEnums
  - CameraDefs Class, [75](#)
- ChunkRegionID
  - Spinnaker::Camera, [430](#)
- ChunkScan3dAxisMax
  - Spinnaker::Camera, [430](#)
- ChunkScan3dAxisMin
  - Spinnaker::Camera, [430](#)
- ChunkScan3dCoordinateOffset
  - Spinnaker::Camera, [430](#)
- ChunkScan3dCoordinateReferenceSelector
  - Spinnaker::Camera, [431](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationX
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationY
  - CameraDefs Class, [76](#)
- CameraDefs Class, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_RotationZ
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationX
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationY
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateReferenceSelector\_TranslationZ
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateReferenceSelectorEnums
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateReferenceValue
  - Spinnaker::Camera, [431](#)
- ChunkScan3dCoordinateScale
  - Spinnaker::Camera, [431](#)
- ChunkScan3dCoordinateSelector
  - Spinnaker::Camera, [431](#)
- ChunkScan3dCoordinateSelector\_CoordinateA
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateSelector\_CoordinateB
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateSelector\_CoordinateC
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateSelectorEnums
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateSystem
  - Spinnaker::Camera, [431](#)
- ChunkScan3dCoordinateSystem\_Cartesian
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateSystem\_Cylindrical
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateSystem\_Spherical
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateSystemEnums
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateSystemReference
  - Spinnaker::Camera, [431](#)
- ChunkScan3dCoordinateSystemReference\_Anchor
  - CameraDefs Class, [77](#)
- ChunkScan3dCoordinateSystemReference\_Transformed
  - CameraDefs Class, [77](#)
- ChunkScan3dCoordinateSystemReferenceEnums
  - CameraDefs Class, [76](#)
- ChunkScan3dCoordinateTransformSelector
  - Spinnaker::Camera, [431](#)
- ChunkScan3dCoordinateTransformSelector\_RotationX
  - CameraDefs Class, [77](#)
- ChunkScan3dCoordinateTransformSelector\_RotationY
  - CameraDefs Class, [77](#)
- ChunkScan3dCoordinateTransformSelector\_RotationZ
  - CameraDefs Class, [77](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationX
  - CameraDefs Class, [77](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationY
  - CameraDefs Class, [77](#)
- ChunkScan3dCoordinateTransformSelector\_TranslationZ
  - CameraDefs Class, [77](#)
- ChunkScan3dCoordinateTransformSelectorEnums



- CameraDefs Class, [77](#)
- ChunkScan3dDistanceUnit
  - Spinnaker::Camera, [432](#)
- ChunkScan3dDistanceUnit\_Inch
  - CameraDefs Class, [77](#)
- ChunkScan3dDistanceUnit\_Millimeter
  - CameraDefs Class, [77](#)
- ChunkScan3dDistanceUnitEnums
  - CameraDefs Class, [77](#)
- ChunkScan3dInvalidDataFlag
  - Spinnaker::Camera, [432](#)
- ChunkScan3dInvalidDataValue
  - Spinnaker::Camera, [432](#)
- ChunkScan3dOutputMode
  - Spinnaker::Camera, [432](#)
- ChunkScan3dOutputMode\_CalibratedABC\_Grid
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_CalibratedABC\_PointCloud
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_CalibratedAC\_Linescan
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_CalibratedAC
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_CalibratedC\_Linescan
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_CalibratedC
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_DisparityC\_Linescan
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_DisparityC
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_RectifiedC\_Linescan
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_RectifiedC
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputMode\_UncalibratedC
  - CameraDefs Class, [78](#)
- ChunkScan3dOutputModeEnums
  - CameraDefs Class, [77](#)
- ChunkScan3dTransformValue
  - Spinnaker::Camera, [432](#)
- ChunkScanLineSelector
  - Spinnaker::Camera, [432](#)
- ChunkSelector
  - Spinnaker::Camera, [432](#)
- ChunkSelector\_BlackLevel
  - CameraDefs Class, [79](#)
- ChunkSelector\_CRC
  - CameraDefs Class, [78](#)
- ChunkSelector\_ExposureEndLineStatusAll
  - CameraDefs Class, [79](#)
- ChunkSelector\_ExposureTime
  - CameraDefs Class, [78](#)
- ChunkSelector\_FrameID
  - CameraDefs Class, [78](#)
- ChunkSelector\_Gain
  - CameraDefs Class, [79](#)
- ChunkSelector\_Height
  - CameraDefs Class, [78](#)
- ChunkSelector\_Image
  - CameraDefs Class, [78](#)
- ChunkSelector\_OffsetX
  - CameraDefs Class, [78](#)
- ChunkSelector\_OffsetY
  - CameraDefs Class, [78](#)
- ChunkSelector\_PixelFormat
  - CameraDefs Class, [79](#)
- ChunkSelector\_SequencerSetActive
  - CameraDefs Class, [79](#)
- ChunkSelector\_SerialData
  - CameraDefs Class, [79](#)
- ChunkSelector\_Timestamp
  - CameraDefs Class, [79](#)
- ChunkSelector\_Width
  - CameraDefs Class, [78](#)
- ChunkSelectorEnums
  - CameraDefs Class, [78](#)
- ChunkSequencerSetActive
  - Spinnaker::Camera, [433](#)
- ChunkSerialData
  - Spinnaker::Camera, [433](#)
- ChunkSerialDataLength
  - Spinnaker::Camera, [433](#)
- ChunkSerialReceiveOverflow
  - Spinnaker::Camera, [433](#)
- ChunkSourceID\_Source0
  - CameraDefs Class, [79](#)
- ChunkSourceID\_Source1
  - CameraDefs Class, [79](#)
- ChunkSourceID\_Source2
  - CameraDefs Class, [79](#)
- ChunkSourceIDEnums
  - CameraDefs Class, [79](#)
- ChunkSourceID
  - Spinnaker::Camera, [433](#)
- ChunkStreamChannelID
  - Spinnaker::Camera, [433](#)
- ChunkTimerSelector
  - Spinnaker::Camera, [433](#)
- ChunkTimerSelector\_Timer0
  - CameraDefs Class, [79](#)
- ChunkTimerSelector\_Timer1
  - CameraDefs Class, [79](#)
- ChunkTimerSelector\_Timer2
  - CameraDefs Class, [79](#)
- ChunkTimerSelectorEnums
  - CameraDefs Class, [79](#)
- ChunkTimerValue
  - Spinnaker::Camera, [433](#)
- ChunkTimestamp
  - Spinnaker::Camera, [434](#)
- ChunkTimestampLatchValue
  - Spinnaker::Camera, [434](#)
- ChunkTransferBlockID
  - Spinnaker::Camera, [434](#)
- ChunkTransferQueueCurrentBlockCount

- Spinnaker::Camera, [434](#)
- ChunkTransferStreamID\_Stream0
  - CameraDefs Class, [79](#)
- ChunkTransferStreamID\_Stream1
  - CameraDefs Class, [79](#)
- ChunkTransferStreamID\_Stream2
  - CameraDefs Class, [79](#)
- ChunkTransferStreamID\_Stream3
  - CameraDefs Class, [79](#)
- ChunkTransferStreamIDEnums
  - CameraDefs Class, [79](#)
- ChunkTransferStreamID
  - Spinnaker::Camera, [434](#)
- ChunkWidth
  - Spinnaker::Camera, [434](#)
- CL
  - Types Enums, [314](#)
- CIConfiguration
  - Spinnaker::Camera, [434](#)
- CIConfiguration\_Base
  - CameraDefs Class, [80](#)
- CIConfiguration\_DualBase
  - CameraDefs Class, [80](#)
- CIConfiguration\_EightyBit
  - CameraDefs Class, [80](#)
- CIConfiguration\_Full
  - CameraDefs Class, [80](#)
- CIConfiguration\_Medium
  - CameraDefs Class, [80](#)
- CIConfigurationEnums
  - CameraDefs Class, [79](#)
- CITimeSlotsCount
  - Spinnaker::Camera, [435](#)
- CITimeSlotsCount\_One
  - CameraDefs Class, [80](#)
- CITimeSlotsCount\_Three
  - CameraDefs Class, [80](#)
- CITimeSlotsCount\_Two
  - CameraDefs Class, [80](#)
- CITimeSlotsCountEnums
  - CameraDefs Class, [80](#)
- Clear
  - Spinnaker::CameraList, [523](#)
  - Spinnaker::ICameraList, [673](#)
  - Spinnaker::IInterfaceList, [701](#)
  - Spinnaker::InterfaceList, [747](#)
- ClearCache
  - Spinnaker::GenApi::CChunkPort, [540](#)
  - Spinnaker::GenApi::CNodeMapFactory, [586](#)
- ClearCaches
  - Spinnaker::GenApi::CChunkAdapter, [530](#)
- ClearXMLCache
  - Spinnaker::GenApi::NodeMap, [780](#)
- Close
  - Spinnaker::Video::SpinVideo, [807](#)
- close
  - Spinnaker::GenApi::IDevFileStreamBase, [681](#)
  - Spinnaker::GenApi::IDevFileStreamBuf, [683](#)
  - Spinnaker::GenApi::ODevFileStreamBase, [786](#)
  - Spinnaker::GenApi::ODevFileStreamBuf, [788](#)
- closeFile
  - Spinnaker::GenApi::FileProtocolAdapter, [643](#)
- ColorProcessingAlgorithm
  - Spinnaker Definitions, [161](#)
- ColorTransformationEnable
  - Spinnaker::Camera, [435](#)
- ColorTransformationSelector
  - Spinnaker::Camera, [435](#)
- ColorTransformationSelector\_RGBtoRGB
  - CameraDefs Class, [80](#)
- ColorTransformationSelector\_RGBtoYUV
  - CameraDefs Class, [80](#)
- ColorTransformationSelectorEnums
  - CameraDefs Class, [80](#)
- ColorTransformationValue
  - Spinnaker::Camera, [435](#)
- ColorTransformationValueSelector
  - Spinnaker::Camera, [435](#)
- ColorTransformationValueSelector\_Gain00
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Gain01
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Gain02
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Gain10
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Gain11
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Gain12
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Gain20
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Gain21
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Gain22
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Offset0
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Offset1
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelector\_Offset2
  - CameraDefs Class, [81](#)
- ColorTransformationValueSelectorEnums
  - CameraDefs Class, [80](#)
- Combine
  - INode Interface, [255](#)
- Command
  - GVCP\_REQUEST\_HEADER, [664](#)
- CommandHeader
  - U3V\_EVENT\_MESSAGE, [847](#)
- CommandId
  - U3V\_COMMAND\_HEADER, [845](#)
- CommandNode, [595](#)
  - Spinnaker::GenApi::CommandNode, [597](#)
- CommandNode Class, [210](#)
  - CCommandRef, [210](#)

- compare
  - Spinnaker::GenICam::gcstring, [655](#)
- Compatibility.h
  - FMT\_I64, [968](#)
- compression
  - Spinnaker::TIFFOption, [824](#)
- compressionLevel
  - Spinnaker::PNGOption, [790](#)
- CompressionMethod
  - Spinnaker::TIFFOption, [824](#)
- CompressionRatio
  - Spinnaker::Camera, [435](#)
- Connect
  - INodeMap Interface, [260](#)
  - Spinnaker::GenApi::NodeMap, [780](#), [781](#)
- Container Class, [211](#)
- ContentType\_Xml
  - NodeMapFactory Class, [285](#)
- ContentType\_ZippedXml
  - NodeMapFactory Class, [285](#)
- Convert
  - Spinnaker::IImage, [687](#)
  - Spinnaker::Image, [708](#), [709](#)
- Counter, [598](#)
  - Spinnaker::GenApi::Counter, [598](#)
- Counter Class, [212](#)
- CounterDelay
  - Spinnaker::Camera, [436](#)
- CounterDuration
  - Spinnaker::Camera, [436](#)
- CounterEventActivation
  - Spinnaker::Camera, [436](#)
- CounterEventActivation\_AnyEdge
  - CameraDefs Class, [81](#)
- CounterEventActivation\_FallingEdge
  - CameraDefs Class, [81](#)
- CounterEventActivation\_LevelHigh
  - CameraDefs Class, [81](#)
- CounterEventActivation\_LevelLow
  - CameraDefs Class, [81](#)
- CounterEventActivation\_RisingEdge
  - CameraDefs Class, [81](#)
- CounterEventActivationEnums
  - CameraDefs Class, [81](#)
- CounterEventSource
  - Spinnaker::Camera, [436](#)
- CounterEventSource\_Counter0End
  - CameraDefs Class, [82](#)
- CounterEventSource\_Counter0Start
  - CameraDefs Class, [82](#)
- CounterEventSource\_Counter1End
  - CameraDefs Class, [82](#)
- CounterEventSource\_Counter1Start
  - CameraDefs Class, [82](#)
- CounterEventSource\_ExposureEnd
  - CameraDefs Class, [82](#)
- CounterEventSource\_ExposureStart
  - CameraDefs Class, [82](#)
- CounterEventSource\_FrameTriggerWait
  - CameraDefs Class, [82](#)
- CounterEventSource\_Line0
  - CameraDefs Class, [81](#)
- CounterEventSource\_Line1
  - CameraDefs Class, [81](#)
- CounterEventSource\_Line2
  - CameraDefs Class, [81](#)
- CounterEventSource\_Line3
  - CameraDefs Class, [81](#)
- CounterEventSource\_LogicBlock0
  - CameraDefs Class, [82](#)
- CounterEventSource\_LogicBlock1
  - CameraDefs Class, [82](#)
- CounterEventSource\_MHzTick
  - CameraDefs Class, [81](#)
- CounterEventSource\_Off
  - CameraDefs Class, [81](#)
- CounterEventSource\_UserOutput0
  - CameraDefs Class, [81](#)
- CounterEventSource\_UserOutput1
  - CameraDefs Class, [81](#)
- CounterEventSource\_UserOutput2
  - CameraDefs Class, [82](#)
- CounterEventSource\_UserOutput3
  - CameraDefs Class, [82](#)
- CounterEventSourceEnums
  - CameraDefs Class, [81](#)
- CounterReset
  - Spinnaker::Camera, [436](#)
- CounterResetActivation
  - Spinnaker::Camera, [436](#)
- CounterResetActivation\_AnyEdge
  - CameraDefs Class, [82](#)
- CounterResetActivation\_FallingEdge
  - CameraDefs Class, [82](#)
- CounterResetActivation\_LevelHigh
  - CameraDefs Class, [82](#)
- CounterResetActivation\_LevelLow
  - CameraDefs Class, [82](#)
- CounterResetActivation\_RisingEdge
  - CameraDefs Class, [82](#)
- CounterResetActivationEnums
  - CameraDefs Class, [82](#)
- CounterResetSource
  - Spinnaker::Camera, [436](#)
- CounterResetSource\_Counter0End
  - CameraDefs Class, [82](#)
- CounterResetSource\_Counter0Start
  - CameraDefs Class, [82](#)
- CounterResetSource\_Counter1End
  - CameraDefs Class, [82](#)
- CounterResetSource\_Counter1Start
  - CameraDefs Class, [82](#)
- CounterResetSource\_ExposureEnd
  - CameraDefs Class, [82](#)
- CounterResetSource\_ExposureStart
  - CameraDefs Class, [82](#)



- CounterResetSource\_FrameTriggerWait
  - CameraDefs Class, [82](#)
- CounterResetSource\_Line0
  - CameraDefs Class, [82](#)
- CounterResetSource\_Line1
  - CameraDefs Class, [82](#)
- CounterResetSource\_Line2
  - CameraDefs Class, [82](#)
- CounterResetSource\_Line3
  - CameraDefs Class, [82](#)
- CounterResetSource\_LogicBlock0
  - CameraDefs Class, [82](#)
- CounterResetSource\_LogicBlock1
  - CameraDefs Class, [82](#)
- CounterResetSource\_Off
  - CameraDefs Class, [82](#)
- CounterResetSource\_UserOutput0
  - CameraDefs Class, [82](#)
- CounterResetSource\_UserOutput1
  - CameraDefs Class, [82](#)
- CounterResetSource\_UserOutput2
  - CameraDefs Class, [82](#)
- CounterResetSource\_UserOutput3
  - CameraDefs Class, [82](#)
- CounterResetSourceEnums
  - CameraDefs Class, [82](#)
- CounterSelector
  - Spinnaker::Camera, [436](#)
- CounterSelector\_Counter0
  - CameraDefs Class, [83](#)
- CounterSelector\_Counter1
  - CameraDefs Class, [83](#)
- CounterSelectorEnums
  - CameraDefs Class, [82](#)
- CounterStatus
  - Spinnaker::Camera, [437](#)
- CounterStatus\_CounterActive
  - CameraDefs Class, [83](#)
- CounterStatus\_CounterCompleted
  - CameraDefs Class, [83](#)
- CounterStatus\_CounterIdle
  - CameraDefs Class, [83](#)
- CounterStatus\_CounterOverflow
  - CameraDefs Class, [83](#)
- CounterStatus\_CounterTriggerWait
  - CameraDefs Class, [83](#)
- CounterStatusEnums
  - CameraDefs Class, [83](#)
- CounterTriggerActivation
  - Spinnaker::Camera, [437](#)
- CounterTriggerActivation\_AnyEdge
  - CameraDefs Class, [83](#)
- CounterTriggerActivation\_FallingEdge
  - CameraDefs Class, [83](#)
- CounterTriggerActivation\_LevelHigh
  - CameraDefs Class, [83](#)
- CounterTriggerActivation\_LevelLow
  - CameraDefs Class, [83](#)
- CounterTriggerActivation\_RisingEdge
  - CameraDefs Class, [83](#)
- CounterTriggerActivationEnums
  - CameraDefs Class, [83](#)
- CounterTriggerSource
  - Spinnaker::Camera, [437](#)
- CounterTriggerSource\_Counter0End
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_Counter0Start
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_Counter1End
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_Counter1Start
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_ExposureEnd
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_ExposureStart
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_FrameTriggerWait
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_Line0
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_Line1
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_Line2
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_Line3
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_LogicBlock0
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_LogicBlock1
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_Off
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_UserOutput0
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_UserOutput1
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_UserOutput2
  - CameraDefs Class, [84](#)
- CounterTriggerSource\_UserOutput3
  - CameraDefs Class, [84](#)
- CounterTriggerSourceEnums
  - CameraDefs Class, [83](#)
- CounterValue
  - Spinnaker::Camera, [437](#)
- CounterValueAtReset
  - Spinnaker::Camera, [437](#)
- Create
  - Spinnaker::Image, [709](#)
- CreateEmptyNodeMap
  - Spinnaker::GenApi::CNodeMapFactory, [586](#)
- CreateNodeDataFromNodeMap
  - Spinnaker::GenApi::CNodeMapFactory, [586](#)
- CreateNodeMap
  - Spinnaker::GenApi::CNodeMapFactory, [587](#)
- CreateShared
  - Spinnaker::Image, [709](#)

- ctDependingNodes
  - Types Enums, [313](#)
- ctInvalidatingChildren
  - Types Enums, [313](#)
- ctParentNodes
  - Types Enums, [313](#)
- ctReadingChildren
  - Types Enums, [313](#)
- ctTerminalNodes
  - Types Enums, [313](#)
- ctWritingChildren
  - Types Enums, [313](#)
- Custom
  - Types Enums, [313](#)
- CxpConnectionSelector
  - Spinnaker::Camera, [437](#)
- CxpConnectionTestErrorCount
  - Spinnaker::Camera, [437](#)
- CxpConnectionTestMode
  - Spinnaker::Camera, [437](#)
- CxpConnectionTestMode\_Mode1
  - CameraDefs Class, [84](#)
- CxpConnectionTestMode\_Off
  - CameraDefs Class, [84](#)
- CxpConnectionTestModeEnums
  - CameraDefs Class, [84](#)
- CxpConnectionTestPacketCount
  - Spinnaker::Camera, [438](#)
- CxpLinkConfiguration
  - Spinnaker::Camera, [438](#)
- CxpLinkConfiguration\_Auto
  - CameraDefs Class, [84](#)
- CxpLinkConfiguration\_CXP1\_X1
  - CameraDefs Class, [84](#)
- CxpLinkConfiguration\_CXP1\_X2
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP1\_X3
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP1\_X4
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP1\_X5
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP1\_X6
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP2\_X1
  - CameraDefs Class, [84](#)
- CxpLinkConfiguration\_CXP2\_X2
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP2\_X3
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP2\_X4
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP2\_X5
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP2\_X6
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP3\_X1
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP3\_X2
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP3\_X3
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP3\_X4
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP3\_X5
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP3\_X6
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP5\_X1
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP5\_X2
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP5\_X3
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP5\_X4
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP5\_X5
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP5\_X6
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP6\_X1
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP6\_X2
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP6\_X3
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP6\_X4
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP6\_X5
  - CameraDefs Class, [85](#)
- CxpLinkConfiguration\_CXP6\_X6
  - CameraDefs Class, [85](#)
- CxpLinkConfigurationEnums
  - CameraDefs Class, [84](#)
- CxpLinkConfigurationPreferred
  - Spinnaker::Camera, [438](#)
- CxpLinkConfigurationPreferred\_CXP1\_X1
  - CameraDefs Class, [85](#)
- CxpLinkConfigurationPreferred\_CXP1\_X2
  - CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP1\_X3
  - CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP1\_X4
  - CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP1\_X5
  - CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP1\_X6
  - CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP2\_X1
  - CameraDefs Class, [85](#)
- CxpLinkConfigurationPreferred\_CXP2\_X2
  - CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP2\_X3
  - CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP2\_X4
  - CameraDefs Class, [86](#)

- CxpLinkConfigurationPreferred\_CXP2\_X5  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP2\_X6  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP3\_X1  
CameraDefs Class, [85](#)
- CxpLinkConfigurationPreferred\_CXP3\_X2  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP3\_X3  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP3\_X4  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP3\_X5  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP3\_X6  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP5\_X1  
CameraDefs Class, [85](#)
- CxpLinkConfigurationPreferred\_CXP5\_X2  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP5\_X3  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP5\_X4  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP5\_X5  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP5\_X6  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP6\_X1  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP6\_X2  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP6\_X3  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP6\_X4  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP6\_X5  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferred\_CXP6\_X6  
CameraDefs Class, [86](#)
- CxpLinkConfigurationPreferredEnums  
CameraDefs Class, [85](#)
- CxpLinkConfigurationStatus  
Spinnaker::Camera, [438](#)
- CxpLinkConfigurationStatus\_CXP1\_X1  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatus\_CXP1\_X2  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatus\_CXP1\_X3  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP1\_X4  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP1\_X5  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP1\_X6  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP2\_X1  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatus\_CXP2\_X2  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatus\_CXP2\_X3  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP2\_X4  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP2\_X5  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP2\_X6  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X1  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatus\_CXP3\_X2  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X3  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X4  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X5  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP3\_X6  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X1  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatus\_CXP5\_X2  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X3  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X4  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X5  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP5\_X6  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X1  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatus\_CXP6\_X2  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X3  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X4  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X5  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_CXP6\_X6  
CameraDefs Class, [87](#)
- CxpLinkConfigurationStatus\_None  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatus\_Pending  
CameraDefs Class, [86](#)
- CxpLinkConfigurationStatusEnums  
CameraDefs Class, [86](#)
- CxpPoCxpAuto  
Spinnaker::Camera, [438](#)
- CxpPoCxpStatus  
Spinnaker::Camera, [438](#)
- CxpPoCxpStatus\_Auto  
CameraDefs Class, [87](#)

- CxpPoCxpStatus\_Off
  - CameraDefs Class, [87](#)
- CxpPoCxpStatus\_Tripped
  - CameraDefs Class, [87](#)
- CxpPoCxpStatusEnums
  - CameraDefs Class, [87](#)
- CxpPoCxpTripReset
  - Spinnaker::Camera, [438](#)
- CxpPoCxpTurnOff
  - Spinnaker::Camera, [439](#)
- DCAM\_CHECKSUM, [614](#)
  - CRCChecksum, [615](#)
- DCAM\_CHUNK\_TRAILER, [615](#)
  - ChunkID, [615](#)
  - ChunkLength, [615](#)
  - InverseChunkLength, [615](#)
- DEFAULT
  - Spinnaker Definitions, [161](#)
- DEFLATE
  - Spinnaker::TIFFOption, [824](#)
- DEPRECATED\_CLASS
  - AVI Recorder Class, [33](#)
- DIRECTIONAL\_FILTER
  - Spinnaker Definitions, [161](#)
- DOLP\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- DOLP\_HEATMAP
  - Spinnaker Definitions, [166](#)
- Data
  - GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_↔ ID, [663](#)
  - GVCP\_EVENTDATA\_REQUEST, [662](#)
- Delnit
  - Spinnaker::CameraBase, [515](#)
  - Spinnaker::ICameraBase, [669](#)
- DecimationHorizontal
  - Spinnaker::Camera, [439](#)
- DecimationHorizontalMode
  - Spinnaker::Camera, [439](#)
- DecimationHorizontalMode\_Discard
  - CameraDefs Class, [87](#)
- DecimationHorizontalModeEnums
  - CameraDefs Class, [87](#)
- DecimationSelector
  - Spinnaker::Camera, [439](#)
- DecimationSelector\_All
  - CameraDefs Class, [88](#)
- DecimationSelector\_Sensor
  - CameraDefs Class, [88](#)
- DecimationSelectorEnums
  - CameraDefs Class, [87](#)
- DecimationVertical
  - Spinnaker::Camera, [439](#)
- DecimationVerticalMode
  - Spinnaker::Camera, [440](#)
- DecimationVerticalMode\_Discard
  - CameraDefs Class, [88](#)
- DecimationVerticalModeEnums
  - CameraDefs Class, [88](#)
- Decreasing
  - Types Enums, [314](#)
- DeepCopy
  - Spinnaker::Image, [687](#)
  - Spinnaker::Image, [709](#), [710](#)
- DefectCorrectStaticEnable
  - Spinnaker::Camera, [440](#)
- DefectCorrectionMode
  - Spinnaker::Camera, [440](#)
- DefectCorrectionMode\_Average
  - CameraDefs Class, [88](#)
- DefectCorrectionMode\_Highlight
  - CameraDefs Class, [88](#)
- DefectCorrectionMode\_Zero
  - CameraDefs Class, [88](#)
- DefectCorrectionModeEnums
  - CameraDefs Class, [88](#)
- DefectTableApply
  - Spinnaker::Camera, [440](#)
- DefectTableCoordinateX
  - Spinnaker::Camera, [440](#)
- DefectTableCoordinateY
  - Spinnaker::Camera, [441](#)
- DefectTableFactoryRestore
  - Spinnaker::Camera, [441](#)
- DefectTableIndex
  - Spinnaker::Camera, [441](#)
- DefectTablePixelCount
  - Spinnaker::Camera, [441](#)
- DefectTableSave
  - Spinnaker::Camera, [441](#)
- Deinterlacing
  - Spinnaker::Camera, [442](#)
- Deinterlacing\_LineDuplication
  - CameraDefs Class, [88](#)
- Deinterlacing\_Off
  - CameraDefs Class, [88](#)
- Deinterlacing\_Weave
  - CameraDefs Class, [88](#)
- DeinterlacingEnums
  - CameraDefs Class, [88](#)
- deleteFile
  - Spinnaker::GenApi::FileProtocolAdapter, [643](#)
- DeliverEventMessage
  - Spinnaker::GenApi::CEventAdapter1394, [549](#)
  - Spinnaker::GenApi::CEventAdapterGEV, [552](#)
  - Spinnaker::GenApi::CEventAdapterU3V, [554](#)
- DeliverMessage
  - Spinnaker::GenApi::CEventAdapter, [547](#)
  - Spinnaker::GenApi::CEventAdapter1394, [549](#)
  - Spinnaker::GenApi::CEventAdapterGEV, [552](#)
  - Spinnaker::GenApi::CEventAdapterGeneric, [551](#)
  - Spinnaker::GenApi::CEventAdapterU3V, [554](#)
- Deregister
  - NodeCallback Class, [282](#)
- DeregisterCallback
  - INode Interface, [255](#)

- Spinnaker::GenApi::Node, [773](#)
- Destroy
  - Spinnaker::GenApi::CNodeCallback, [581](#)
  - Spinnaker::GenApi::Function\_NodeCallback, [653](#)
  - Spinnaker::GenApi::Member\_NodeCallback, [769](#)
  - Spinnaker::GenApi::NodeMap, [781](#)
- DetachBuffer
  - Spinnaker::GenApi::CChunkAdapter, [530](#)
- DetachChunk
  - Spinnaker::GenApi::CChunkPort, [540](#)
- DetachEvent
  - Spinnaker::GenApi::CEventPort, [556](#)
- DetachNode
  - Spinnaker::GenApi::CEventPort, [556](#)
- DetachNodeMap
  - Spinnaker::GenApi::CChunkAdapter, [531](#)
  - Spinnaker::GenApi::CEventAdapter, [547](#)
- DetachPort
  - Spinnaker::GenApi::CChunkPort, [541](#)
- DeviceAccessStatus
  - Spinnaker::TransportLayerDevice, [827](#)
  - Spinnaker::TransportLayerInterface, [834](#)
- DeviceAccessStatus\_NoAccess
  - TransportLayerDefs Class, [175](#)
- DeviceAccessStatus\_ReadOnly
  - TransportLayerDefs Class, [175](#)
- DeviceAccessStatus\_ReadWrite
  - TransportLayerDefs Class, [175](#)
- DeviceAccessStatus\_Unknown
  - TransportLayerDefs Class, [175](#)
- DeviceAccessStatusEnum
  - TransportLayerDefs Class, [175](#)
- DeviceAddress
  - Spinnaker::ActionCommandResult, [375](#)
- DeviceCharacterSet
  - Spinnaker::Camera, [442](#)
- DeviceCharacterSet\_ASCII
  - CameraDefs Class, [89](#)
- DeviceCharacterSet\_UTF8
  - CameraDefs Class, [89](#)
- DeviceCharacterSetEnums
  - CameraDefs Class, [88](#)
- DeviceClockFrequency
  - Spinnaker::Camera, [442](#)
- DeviceClockSelector
  - Spinnaker::Camera, [442](#)
- DeviceClockSelector\_CameraLink
  - CameraDefs Class, [89](#)
- DeviceClockSelector\_Sensor
  - CameraDefs Class, [89](#)
- DeviceClockSelector\_SensorDigitization
  - CameraDefs Class, [89](#)
- DeviceClockSelectorEnums
  - CameraDefs Class, [89](#)
- DeviceConnectionSelector
  - Spinnaker::Camera, [442](#)
- DeviceConnectionSpeed
  - Spinnaker::Camera, [442](#)
- DeviceConnectionStatus
  - Spinnaker::Camera, [442](#)
- DeviceConnectionStatus\_Active
  - CameraDefs Class, [89](#)
- DeviceConnectionStatus\_Inactive
  - CameraDefs Class, [89](#)
- DeviceConnectionStatusEnums
  - CameraDefs Class, [89](#)
- DeviceCount
  - Spinnaker::TransportLayerInterface, [834](#)
- DeviceCurrentSpeed
  - Spinnaker::TransportLayerDevice, [827](#)
- DeviceCurrentSpeed\_FullSpeed
  - TransportLayerDefs Class, [176](#)
- DeviceCurrentSpeed\_HighSpeed
  - TransportLayerDefs Class, [176](#)
- DeviceCurrentSpeed\_LowSpeed
  - TransportLayerDefs Class, [176](#)
- DeviceCurrentSpeed\_SuperSpeed
  - TransportLayerDefs Class, [176](#)
- DeviceCurrentSpeed\_UnknownSpeed
  - TransportLayerDefs Class, [176](#)
- DeviceCurrentSpeedEnum
  - TransportLayerDefs Class, [175](#)
- DeviceDisplayName
  - Spinnaker::TransportLayerDevice, [827](#)
- DeviceDriverVersion
  - Spinnaker::TransportLayerDevice, [827](#)
- DeviceEndianessMechanism
  - Spinnaker::TransportLayerDevice, [827](#)
- DeviceEndianessMechanism\_Legacy
  - TransportLayerDefs Class, [176](#)
- DeviceEndianessMechanism\_Standard
  - TransportLayerDefs Class, [176](#)
- DeviceEndianessMechanismEnum
  - TransportLayerDefs Class, [176](#)
- DeviceEvent, [615](#)
  - Spinnaker::DeviceEvent, [616](#)
- DeviceEvent Class, [139](#)
- DeviceEventChannelCount
  - Spinnaker::Camera, [443](#)
- DeviceFamilyName
  - Spinnaker::Camera, [443](#)
- DeviceFeaturePersistenceEnd
  - Spinnaker::Camera, [443](#)
- DeviceFeaturePersistenceStart
  - Spinnaker::Camera, [443](#)
- DeviceFirmwareVersion
  - Spinnaker::Camera, [443](#)
- DeviceGenCPVersionMajor
  - Spinnaker::Camera, [443](#)
- DeviceGenCPVersionMinor
  - Spinnaker::Camera, [443](#)
- DeviceID
  - Spinnaker::Camera, [444](#)
  - Spinnaker::TransportLayerDevice, [827](#)
  - Spinnaker::TransportLayerInterface, [834](#)
- DeviceIndicatorMode

- Spinnaker::Camera, [444](#)
- DeviceIndicatorMode\_Active
  - CameraDefs Class, [89](#)
- DeviceIndicatorMode\_ErrorStatus
  - CameraDefs Class, [89](#)
- DeviceIndicatorMode\_Inactive
  - CameraDefs Class, [89](#)
- DeviceIndicatorModeEnums
  - CameraDefs Class, [89](#)
- DeviceInstanceId
  - Spinnaker::TransportLayerDevice, [828](#)
- DevicesUpdater
  - Spinnaker::TransportLayerDevice, [828](#)
- DeviceLinkBandwidthReserve
  - Spinnaker::Camera, [444](#)
- DeviceLinkCommandTimeout
  - Spinnaker::Camera, [444](#)
- DeviceLinkConnectionCount
  - Spinnaker::Camera, [444](#)
- DeviceLinkCurrentThroughput
  - Spinnaker::Camera, [444](#)
- DeviceLinkHeartbeatMode
  - Spinnaker::Camera, [444](#)
- DeviceLinkHeartbeatMode\_Off
  - CameraDefs Class, [90](#)
- DeviceLinkHeartbeatMode\_On
  - CameraDefs Class, [90](#)
- DeviceLinkHeartbeatModeEnums
  - CameraDefs Class, [89](#)
- DeviceLinkHeartbeatTimeout
  - Spinnaker::Camera, [445](#)
- DeviceLinkSelector
  - Spinnaker::Camera, [445](#)
- DeviceLinkSpeed
  - Spinnaker::Camera, [445](#)
  - Spinnaker::TransportLayerDevice, [828](#)
- DeviceLinkThroughputLimit
  - Spinnaker::Camera, [445](#)
- DeviceLinkThroughputLimitMode
  - Spinnaker::Camera, [445](#)
- DeviceLinkThroughputLimitMode\_Off
  - CameraDefs Class, [90](#)
- DeviceLinkThroughputLimitMode\_On
  - CameraDefs Class, [90](#)
- DeviceLinkThroughputLimitModeEnums
  - CameraDefs Class, [90](#)
- DeviceManifestEntrySelector
  - Spinnaker::Camera, [445](#)
- DeviceManifestPrimaryURL
  - Spinnaker::Camera, [446](#)
- DeviceManifestSchemaMajorVersion
  - Spinnaker::Camera, [446](#)
- DeviceManifestSchemaMinorVersion
  - Spinnaker::Camera, [446](#)
- DeviceManifestSecondaryURL
  - Spinnaker::Camera, [446](#)
- DeviceManifestXMLMajorVersion
  - Spinnaker::Camera, [446](#)
- DeviceManifestXMLMinorVersion
  - Spinnaker::Camera, [446](#)
- DeviceManifestXMLSubMinorVersion
  - Spinnaker::Camera, [446](#)
- DeviceManufacturerInfo
  - Spinnaker::Camera, [446](#)
- DeviceMaxThroughput
  - Spinnaker::Camera, [447](#)
- DeviceModelName
  - Spinnaker::Camera, [447](#)
  - Spinnaker::TransportLayerDevice, [828](#)
  - Spinnaker::TransportLayerInterface, [835](#)
- DeviceMulticastMonitorMode
  - Spinnaker::TransportLayerDevice, [828](#)
- DevicePowerSupplySelector
  - Spinnaker::Camera, [447](#)
- DevicePowerSupplySelector\_External
  - CameraDefs Class, [90](#)
- DevicePowerSupplySelectorEnums
  - CameraDefs Class, [90](#)
- DeviceRegistersCheck
  - Spinnaker::Camera, [447](#)
- DeviceRegistersEndianness
  - Spinnaker::Camera, [447](#)
- DeviceRegistersEndianness\_Big
  - CameraDefs Class, [90](#)
- DeviceRegistersEndianness\_Little
  - CameraDefs Class, [90](#)
- DeviceRegistersEndiannessEnums
  - CameraDefs Class, [90](#)
- DeviceRegistersStreamingEnd
  - Spinnaker::Camera, [447](#)
- DeviceRegistersStreamingStart
  - Spinnaker::Camera, [448](#)
- DeviceRegistersValid
  - Spinnaker::Camera, [448](#)
- DeviceReset
  - Spinnaker::Camera, [448](#)
- DeviceSFNCVersionMajor
  - Spinnaker::Camera, [449](#)
- DeviceSFNCVersionMinor
  - Spinnaker::Camera, [449](#)
- DeviceSFNCVersionSubMinor
  - Spinnaker::Camera, [449](#)
- DeviceScanType
  - Spinnaker::Camera, [448](#)
- DeviceScanType\_Areascan
  - CameraDefs Class, [90](#)
- DeviceScanTypeEnums
  - CameraDefs Class, [90](#)
- DeviceSelector
  - Spinnaker::TransportLayerInterface, [835](#)
- DeviceSerialNumber
  - Spinnaker::Camera, [448](#)
  - Spinnaker::TransportLayerDevice, [828](#)
- DeviceSerialPortBaudRate
  - Spinnaker::Camera, [448](#)
- DeviceSerialPortBaudRate\_Baud115200



- CameraDefs Class, [91](#)
- DeviceSerialPortBaudRate\_Baud19200
  - CameraDefs Class, [91](#)
- DeviceSerialPortBaudRate\_Baud230400
  - CameraDefs Class, [91](#)
- DeviceSerialPortBaudRate\_Baud38400
  - CameraDefs Class, [91](#)
- DeviceSerialPortBaudRate\_Baud460800
  - CameraDefs Class, [91](#)
- DeviceSerialPortBaudRate\_Baud57600
  - CameraDefs Class, [91](#)
- DeviceSerialPortBaudRate\_Baud921600
  - CameraDefs Class, [91](#)
- DeviceSerialPortBaudRate\_Baud9600
  - CameraDefs Class, [91](#)
- DeviceSerialPortBaudRateEnums
  - CameraDefs Class, [90](#)
- DeviceSerialPortSelector
  - Spinnaker::Camera, [448](#)
- DeviceSerialPortSelector\_CameraLink
  - CameraDefs Class, [91](#)
- DeviceSerialPortSelectorEnums
  - CameraDefs Class, [91](#)
- DeviceStreamChannelCount
  - Spinnaker::Camera, [449](#)
- DeviceStreamChannelEndianness
  - Spinnaker::Camera, [449](#)
- DeviceStreamChannelEndianness\_Big
  - CameraDefs Class, [91](#)
- DeviceStreamChannelEndianness\_Little
  - CameraDefs Class, [91](#)
- DeviceStreamChannelEndiannessEnums
  - CameraDefs Class, [91](#)
- DeviceStreamChannelLink
  - Spinnaker::Camera, [449](#)
- DeviceStreamChannelPacketSize
  - Spinnaker::Camera, [449](#)
- DeviceStreamChannelSelector
  - Spinnaker::Camera, [450](#)
- DeviceStreamChannelType
  - Spinnaker::Camera, [450](#)
- DeviceStreamChannelType\_Receiver
  - CameraDefs Class, [91](#)
- DeviceStreamChannelType\_Transmitter
  - CameraDefs Class, [91](#)
- DeviceStreamChannelTypeEnums
  - CameraDefs Class, [91](#)
- DeviceTLType
  - Spinnaker::Camera, [450](#)
- DeviceTLType\_CameraLink
  - CameraDefs Class, [93](#)
- DeviceTLType\_CameraLinkHS
  - CameraDefs Class, [93](#)
- DeviceTLType\_CoaxPress
  - CameraDefs Class, [93](#)
- DeviceTLType\_Custom
  - CameraDefs Class, [93](#)
- DeviceTLType\_GigEVision
  - CameraDefs Class, [93](#)
- DeviceTLType\_USB3Vision
  - CameraDefs Class, [93](#)
- DeviceTLTypeEnums
  - CameraDefs Class, [93](#)
- DeviceTLVersionMajor
  - Spinnaker::Camera, [450](#)
- DeviceTLVersionMinor
  - Spinnaker::Camera, [451](#)
- DeviceTLVersionSubMinor
  - Spinnaker::Camera, [451](#)
- DeviceTapGeometry
  - Spinnaker::Camera, [450](#)
- DeviceTapGeometry\_Geometry\_10X\_1Y
  - CameraDefs Class, [93](#)
- DeviceTapGeometry\_Geometry\_10X
  - CameraDefs Class, [93](#)
- DeviceTapGeometry\_Geometry\_1X10
  - CameraDefs Class, [93](#)
- DeviceTapGeometry\_Geometry\_1X10\_1Y
  - CameraDefs Class, [93](#)
- DeviceTapGeometry\_Geometry\_1X2
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X2\_1Y2
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X2\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X2\_2YE
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X3
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X3\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X4
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X4\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X8
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X8\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X\_1Y2
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X\_2YE
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_1X
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X2
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X2\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X2E\_1YGeometry↵
  - \_2X2M\_1Y
    - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X2E\_2YE
  - CameraDefs Class, [92](#)

- DeviceTapGeometry\_Geometry\_2X2E
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X2M
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X\_1Y2Geometry\_2X\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X\_2YE
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2XE\_1Y2
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2XE\_2YE
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2XM\_1Y2
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2XM\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2XM\_2YE
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2XE
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2XM
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_2X
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_3X\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_3X
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_4X2
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_4X2\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_4X2E\_1Y
  - CameraDefs Class, [93](#)
- DeviceTapGeometry\_Geometry\_4X2E
  - CameraDefs Class, [93](#)
- DeviceTapGeometry\_Geometry\_4X\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_4X
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_8X\_1Y
  - CameraDefs Class, [92](#)
- DeviceTapGeometry\_Geometry\_8X
  - CameraDefs Class, [92](#)
- DeviceTapGeometryEnums
  - CameraDefs Class, [91](#)
- DeviceTemperature
  - Spinnaker::Camera, [450](#)
- DeviceTemperatureSelector
  - Spinnaker::Camera, [450](#)
- DeviceTemperatureSelector\_Sensor
  - CameraDefs Class, [93](#)
- DeviceTemperatureSelectorEnums
  - CameraDefs Class, [93](#)
- DeviceType
  - Spinnaker::Camera, [451](#)
  - Spinnaker::TransportLayerDevice, [828](#)
- DeviceType\_CLHS
  - TransportLayerDefs Class, [176](#)
- DeviceType\_CXP
  - TransportLayerDefs Class, [176](#)
- DeviceType\_CL
  - TransportLayerDefs Class, [176](#)
- DeviceType\_Custom
  - TransportLayerDefs Class, [176](#)
- DeviceType\_ETHERNET
  - TransportLayerDefs Class, [176](#)
- DeviceType\_GEV
  - TransportLayerDefs Class, [176](#)
- DeviceType\_IIDC
  - TransportLayerDefs Class, [176](#)
- DeviceType\_Mixed
  - TransportLayerDefs Class, [176](#)
- DeviceType\_PCI
  - TransportLayerDefs Class, [176](#)
- DeviceType\_Peripheral
  - CameraDefs Class, [93](#)
- DeviceType\_Receiver
  - CameraDefs Class, [93](#)
- DeviceType\_Transceiver
  - CameraDefs Class, [93](#)
- DeviceType\_Transmitter
  - CameraDefs Class, [93](#)
- DeviceType\_U3V
  - TransportLayerDefs Class, [176](#)
- DeviceType\_UVC
  - TransportLayerDefs Class, [176](#)
- DeviceTypeEnum
  - TransportLayerDefs Class, [176](#)
- DeviceTypeEnums
  - CameraDefs Class, [93](#)
- DeviceU3VProtocol
  - Spinnaker::TransportLayerDevice, [828](#)
- DeviceUnlock
  - Spinnaker::TransportLayerInterface, [835](#)
- DeviceUpdateList
  - Spinnaker::TransportLayerInterface, [835](#)
- DeviceUptime
  - Spinnaker::Camera, [451](#)
- DeviceUserID
  - Spinnaker::Camera, [451](#)
  - Spinnaker::TransportLayerDevice, [829](#)
- DeviceVendorName
  - Spinnaker::Camera, [451](#)
  - Spinnaker::TransportLayerDevice, [829](#)
  - Spinnaker::TransportLayerInterface, [835](#)
- DeviceVersion
  - Spinnaker::Camera, [451](#)
  - Spinnaker::TransportLayerDevice, [829](#)
- DisableAll
  - Spinnaker::ImageStatistics, [694](#)
  - Spinnaker::ImageStatistics, [730](#)
- DiscoverMaxPacketSize



- Spinnaker::CameraBase, [516](#)
- Spinnaker::ICameraBase, [669](#)
- doc/Doxygen/spindocs/Licensing.dox, [851](#)
- doc/Doxygen/spindocs/MainPage.dox, [851](#)
- DoesEnvironmentVariableExist
  - GCUtilities Utility, [232](#)
- double\_autovector\_t, [618](#)
  - Spinnaker::GenApi::double\_autovector\_t, [618](#)
- EAccessMode
  - Types Enums, [311](#)
- EAccessModeClass, [619](#)
- ECacheUsage\_t
  - NodeMapFactory Class, [284](#)
- ECachingMode
  - Types Enums, [311](#)
- ECachingModeClass, [620](#)
- ECallbackType
  - NodeCallback Class, [282](#)
- EContentType\_t
  - NodeMapFactory Class, [284](#)
- EDGE\_SENSING
  - Spinnaker Definitions, [161](#)
- EDisplayNotation
  - Types Enums, [311](#)
- EDisplayNotationClass, [620](#)
- EEndianess
  - Types Enums, [311](#)
- EEndianessClass, [621](#)
- EGenApiSchemaVersion
  - Types Enums, [311](#)
- EGenApiSchemaVersionClass, [622](#)
- ElncMode
  - Types Enums, [312](#)
- EInputDirection
  - Types Enums, [312](#)
- EInputDirectionClass, [623](#)
- EInterfaceType
  - Types Enums, [312](#)
- ELinkType
  - Types Enums, [312](#)
- ENameSpace
  - Types Enums, [313](#)
- ENameSpaceClass, [624](#)
- ERepresentation
  - Types Enums, [313](#)
- ERepresentationClass, [632](#)
- ESign
  - Types Enums, [313](#)
- ESignClass, [632](#)
- ESlope
  - Types Enums, [313](#)
- ESlopeClass, [633](#)
- EStandardNameSpace
  - Types Enums, [314](#)
- EStandardNameSpaceClass, [634](#)
- EVENT\_TIMEOUT\_INFINITY
  - Spinnaker Headers, [155](#)
- EVENT\_TIMEOUT\_NONE
  - Spinnaker Headers, [155](#)
- EVisibility
  - Types Enums, [314](#)
- EVisibilityClass, [637](#)
- EXMLValidation
  - Types Enums, [314](#)
- EXPAND\_TO\_STRINGISE
  - GCUtilities.h, [1004](#)
- EYesNo
  - Types Enums, [314](#)
- EYesNoClass, [641](#)
- EatComments
  - Spinnaker GenApi Classes, [197](#)
- empty
  - Spinnaker::GenICam::gcstring, [655](#)
- EnableAll
  - Spinnaker::ImageStatistics, [694](#)
  - Spinnaker::ImageStatistics, [730](#)
- EnableGreyOnly
  - Spinnaker::ImageStatistics, [694](#)
  - Spinnaker::ImageStatistics, [730](#)
- EnableHSLOnly
  - Spinnaker::ImageStatistics, [695](#)
  - Spinnaker::ImageStatistics, [730](#)
- EnableRGBOnly
  - Spinnaker::ImageStatistics, [695](#)
  - Spinnaker::ImageStatistics, [730](#)
- EncoderDivider
  - Spinnaker::Camera, [452](#)
- EncoderMode
  - Spinnaker::Camera, [452](#)
- EncoderMode\_FourPhase
  - CameraDefs Class, [94](#)
- EncoderMode\_HighResolution
  - CameraDefs Class, [94](#)
- EncoderModeEnums
  - CameraDefs Class, [93](#)
- EncoderOutputMode
  - Spinnaker::Camera, [452](#)
- EncoderOutputMode\_DirectionDown
  - CameraDefs Class, [94](#)
- EncoderOutputMode\_DirectionUp
  - CameraDefs Class, [94](#)
- EncoderOutputMode\_Motion
  - CameraDefs Class, [94](#)
- EncoderOutputMode\_Off
  - CameraDefs Class, [94](#)
- EncoderOutputMode\_PositionDown
  - CameraDefs Class, [94](#)
- EncoderOutputMode\_PositionUp
  - CameraDefs Class, [94](#)
- EncoderOutputModeEnums
  - CameraDefs Class, [94](#)
- EncoderReset
  - Spinnaker::Camera, [452](#)
- EncoderResetActivation
  - Spinnaker::Camera, [452](#)
- EncoderResetActivation\_AnyEdge

- CameraDefs Class, [94](#)
- EncoderResetActivation\_FallingEdge
  - CameraDefs Class, [94](#)
- EncoderResetActivation\_LevelHigh
  - CameraDefs Class, [94](#)
- EncoderResetActivation\_LevelLow
  - CameraDefs Class, [94](#)
- EncoderResetActivation\_RisingEdge
  - CameraDefs Class, [94](#)
- EncoderResetActivationEnums
  - CameraDefs Class, [94](#)
- EncoderResetSource
  - Spinnaker::Camera, [452](#)
- EncoderResetSource\_AcquisitionEnd
  - CameraDefs Class, [95](#)
- EncoderResetSource\_AcquisitionStart
  - CameraDefs Class, [95](#)
- EncoderResetSource\_AcquisitionTrigger
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Action0
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Action1
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Action2
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Counter0End
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Counter0Start
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Counter1End
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Counter1Start
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Counter2End
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Counter2Start
  - CameraDefs Class, [95](#)
- EncoderResetSource\_ExposureEnd
  - CameraDefs Class, [95](#)
- EncoderResetSource\_ExposureStart
  - CameraDefs Class, [95](#)
- EncoderResetSource\_FrameEnd
  - CameraDefs Class, [95](#)
- EncoderResetSource\_FrameStart
  - CameraDefs Class, [95](#)
- EncoderResetSource\_FrameTrigger
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Line0
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Line1
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Line2
  - CameraDefs Class, [95](#)
- EncoderResetSource\_LinkTrigger0
  - CameraDefs Class, [95](#)
- EncoderResetSource\_LinkTrigger1
  - CameraDefs Class, [95](#)
- EncoderResetSource\_LinkTrigger2
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Off
  - CameraDefs Class, [95](#)
- EncoderResetSource\_SoftwareSignal0
  - CameraDefs Class, [95](#)
- EncoderResetSource\_SoftwareSignal1
  - CameraDefs Class, [95](#)
- EncoderResetSource\_SoftwareSignal2
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Timer0End
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Timer0Start
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Timer1End
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Timer1Start
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Timer2End
  - CameraDefs Class, [95](#)
- EncoderResetSource\_Timer2Start
  - CameraDefs Class, [95](#)
- EncoderResetSource\_UserOutput0
  - CameraDefs Class, [95](#)
- EncoderResetSource\_UserOutput1
  - CameraDefs Class, [95](#)
- EncoderResetSource\_UserOutput2
  - CameraDefs Class, [95](#)
- EncoderResetSourceEnums
  - CameraDefs Class, [94](#)
- EncoderSelector
  - Spinnaker::Camera, [452](#)
- EncoderSelector\_Encoder0
  - CameraDefs Class, [96](#)
- EncoderSelector\_Encoder1
  - CameraDefs Class, [96](#)
- EncoderSelector\_Encoder2
  - CameraDefs Class, [96](#)
- EncoderSelectorEnums
  - CameraDefs Class, [95](#)
- EncoderSourceA\_Line0
  - CameraDefs Class, [96](#)
- EncoderSourceA\_Line1
  - CameraDefs Class, [96](#)
- EncoderSourceA\_Line2
  - CameraDefs Class, [96](#)
- EncoderSourceA\_Off
  - CameraDefs Class, [96](#)
- EncoderSourceAEnums
  - CameraDefs Class, [96](#)
- EncoderSourceB\_Line0
  - CameraDefs Class, [96](#)
- EncoderSourceB\_Line1
  - CameraDefs Class, [96](#)
- EncoderSourceB\_Line2
  - CameraDefs Class, [96](#)
- EncoderSourceB\_Off
  - CameraDefs Class, [96](#)
- EncoderSourceBEnums

- CameraDefs Class, [96](#)
- EncoderSourceA
  - Spinnaker::Camera, [453](#)
- EncoderSourceB
  - Spinnaker::Camera, [453](#)
- EncoderStatus
  - Spinnaker::Camera, [453](#)
- EncoderStatus\_EncoderDown
  - CameraDefs Class, [96](#)
- EncoderStatus\_EncoderIdle
  - CameraDefs Class, [96](#)
- EncoderStatus\_EncoderStatic
  - CameraDefs Class, [96](#)
- EncoderStatus\_EncoderUp
  - CameraDefs Class, [96](#)
- EncoderStatusEnums
  - CameraDefs Class, [96](#)
- EncoderTimeout
  - Spinnaker::Camera, [453](#)
- EncoderValue
  - Spinnaker::Camera, [453](#)
- EncoderValueAtReset
  - Spinnaker::Camera, [453](#)
- EndAcquisition
  - Spinnaker::CameraBase, [516](#)
  - Spinnaker::ICameraBase, [669](#)
- EnumClasses Class, [213](#)
- EnumEntryNode, [625](#)
  - Spinnaker::GenApi::EnumEntryNode, [627](#)
- EnumEntryNode Class, [215](#)
  - CEnumEntryRef, [215](#)
- EnumNode, [628](#)
  - Spinnaker::GenApi::EnumNode, [630](#)
- EnumNode Class, [216](#)
  - CEnumerationRef, [216](#)
- EnumNodeT Class, [217](#)
- EnumerationCount
  - Spinnaker::Camera, [453](#)
- Error
  - Spinnaker Definitions, [161](#)
- Event, [635](#)
  - GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_↔ ID, [663](#)
  - GVCP\_EVENTDATA\_REQUEST, [662](#)
  - Spinnaker::Event, [636](#)
- Event Class, [140](#)
- EventAcquisitionEnd
  - Spinnaker::Camera, [454](#)
- EventAcquisitionEndFrameID
  - Spinnaker::Camera, [454](#)
- EventAcquisitionEndTimestamp
  - Spinnaker::Camera, [454](#)
- EventAcquisitionError
  - Spinnaker::Camera, [454](#)
- EventAcquisitionErrorFrameID
  - Spinnaker::Camera, [454](#)
- EventAcquisitionErrorTimestamp
  - Spinnaker::Camera, [454](#)
- EventAcquisitionStart
  - Spinnaker::Camera, [454](#)
- EventAcquisitionStartFrameID
  - Spinnaker::Camera, [454](#)
- EventAcquisitionStartTimestamp
  - Spinnaker::Camera, [455](#)
- EventAcquisitionTransferEnd
  - Spinnaker::Camera, [455](#)
- EventAcquisitionTransferEndFrameID
  - Spinnaker::Camera, [455](#)
- EventAcquisitionTransferEndTimestamp
  - Spinnaker::Camera, [455](#)
- EventAcquisitionTransferStart
  - Spinnaker::Camera, [455](#)
- EventAcquisitionTransferStartFrameID
  - Spinnaker::Camera, [455](#)
- EventAcquisitionTransferStartTimestamp
  - Spinnaker::Camera, [455](#)
- EventAcquisitionTrigger
  - Spinnaker::Camera, [456](#)
- EventAcquisitionTriggerFrameID
  - Spinnaker::Camera, [456](#)
- EventAcquisitionTriggerTimestamp
  - Spinnaker::Camera, [456](#)
- EventActionLate
  - Spinnaker::Camera, [456](#)
- EventActionLateFrameID
  - Spinnaker::Camera, [456](#)
- EventActionLateTimestamp
  - Spinnaker::Camera, [456](#)
- EventAdapter Class, [218](#)
- EventAdapter1394 Class, [219](#)
- EventAdapterGEV Class, [221](#)
- EventAdapterGeneric Class, [220](#)
- EventAdapterU3V Class, [222](#)
- EventCounter0End
  - Spinnaker::Camera, [456](#)
- EventCounter0EndFrameID
  - Spinnaker::Camera, [456](#)
- EventCounter0EndTimestamp
  - Spinnaker::Camera, [457](#)
- EventCounter0Start
  - Spinnaker::Camera, [457](#)
- EventCounter0StartFrameID
  - Spinnaker::Camera, [457](#)
- EventCounter0StartTimestamp
  - Spinnaker::Camera, [457](#)
- EventCounter1End
  - Spinnaker::Camera, [457](#)
- EventCounter1EndFrameID
  - Spinnaker::Camera, [457](#)
- EventCounter1EndTimestamp
  - Spinnaker::Camera, [457](#)
- EventCounter1Start
  - Spinnaker::Camera, [457](#)
- EventCounter1StartFrameID
  - Spinnaker::Camera, [458](#)
- EventCounter1StartTimestamp
  - Spinnaker::Camera, [458](#)

- Spinnaker::Camera, [458](#)
- EventData
  - U3V\_EVENT\_MESSAGE, [847](#)
- EventEncoder0Restarted
  - Spinnaker::Camera, [458](#)
- EventEncoder0RestartedFrameID
  - Spinnaker::Camera, [458](#)
- EventEncoder0RestartedTimestamp
  - Spinnaker::Camera, [458](#)
- EventEncoder0Stopped
  - Spinnaker::Camera, [458](#)
- EventEncoder0StoppedFrameID
  - Spinnaker::Camera, [458](#)
- EventEncoder0StoppedTimestamp
  - Spinnaker::Camera, [458](#)
- EventEncoder1Restarted
  - Spinnaker::Camera, [459](#)
- EventEncoder1RestartedFrameID
  - Spinnaker::Camera, [459](#)
- EventEncoder1RestartedTimestamp
  - Spinnaker::Camera, [459](#)
- EventEncoder1Stopped
  - Spinnaker::Camera, [459](#)
- EventEncoder1StoppedFrameID
  - Spinnaker::Camera, [459](#)
- EventEncoder1StoppedTimestamp
  - Spinnaker::Camera, [459](#)
- EventError
  - Spinnaker::Camera, [459](#)
- EventErrorCode
  - Spinnaker::Camera, [459](#)
- EventErrorFrameID
  - Spinnaker::Camera, [460](#)
- EventErrorTimestamp
  - Spinnaker::Camera, [460](#)
- EventExposureEnd
  - Spinnaker::Camera, [460](#)
- EventExposureEndFrameID
  - Spinnaker::Camera, [460](#)
- EventExposureEndTimestamp
  - Spinnaker::Camera, [460](#)
- EventExposureStart
  - Spinnaker::Camera, [460](#)
- EventExposureStartFrameID
  - Spinnaker::Camera, [460](#)
- EventExposureStartTimestamp
  - Spinnaker::Camera, [460](#)
- EventFrameBurstEnd
  - Spinnaker::Camera, [461](#)
- EventFrameBurstEndFrameID
  - Spinnaker::Camera, [461](#)
- EventFrameBurstEndTimestamp
  - Spinnaker::Camera, [461](#)
- EventFrameBurstStart
  - Spinnaker::Camera, [461](#)
- EventFrameBurstStartFrameID
  - Spinnaker::Camera, [461](#)
- EventFrameBurstStartTimestamp
  - Spinnaker::Camera, [461](#)
- EventFrameEnd
  - Spinnaker::Camera, [461](#)
- EventFrameEndFrameID
  - Spinnaker::Camera, [461](#)
- EventFrameEndTimestamp
  - Spinnaker::Camera, [462](#)
- EventFrameStart
  - Spinnaker::Camera, [462](#)
- EventFrameStartFrameID
  - Spinnaker::Camera, [462](#)
- EventFrameStartTimestamp
  - Spinnaker::Camera, [462](#)
- EventFrameTransferEnd
  - Spinnaker::Camera, [462](#)
- EventFrameTransferEndFrameID
  - Spinnaker::Camera, [462](#)
- EventFrameTransferEndTimestamp
  - Spinnaker::Camera, [462](#)
- EventFrameTransferStart
  - Spinnaker::Camera, [462](#)
- EventFrameTransferStartFrameID
  - Spinnaker::Camera, [463](#)
- EventFrameTransferStartTimestamp
  - Spinnaker::Camera, [463](#)
- EventFrameTrigger
  - Spinnaker::Camera, [463](#)
- EventFrameTriggerFrameID
  - Spinnaker::Camera, [463](#)
- EventFrameTriggerTimestamp
  - Spinnaker::Camera, [463](#)
- EventId
  - GVCP\_EVENT\_ITEM\_BASIC, [659](#)
  - GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [660](#)
  - GVCP\_EVENT\_ITEM, [658](#)
  - U3V\_EVENT\_DATA, [846](#)
- EventLine0AnyEdge
  - Spinnaker::Camera, [463](#)
- EventLine0AnyEdgeFrameID
  - Spinnaker::Camera, [463](#)
- EventLine0AnyEdgeTimestamp
  - Spinnaker::Camera, [464](#)
- EventLine0FallingEdge
  - Spinnaker::Camera, [464](#)
- EventLine0FallingEdgeFrameID
  - Spinnaker::Camera, [464](#)
- EventLine0FallingEdgeTimestamp
  - Spinnaker::Camera, [464](#)
- EventLine0RisingEdge
  - Spinnaker::Camera, [464](#)
- EventLine0RisingEdgeFrameID
  - Spinnaker::Camera, [464](#)
- EventLine0RisingEdgeTimestamp
  - Spinnaker::Camera, [464](#)
- EventLine1AnyEdge
  - Spinnaker::Camera, [464](#)
- EventLine1AnyEdgeFrameID
  - Spinnaker::Camera, [465](#)

- EventLine1AnyEdgeTimestamp
  - Spinnaker::Camera, [465](#)
- EventLine1FallingEdge
  - Spinnaker::Camera, [465](#)
- EventLine1FallingEdgeFrameID
  - Spinnaker::Camera, [465](#)
- EventLine1FallingEdgeTimestamp
  - Spinnaker::Camera, [465](#)
- EventLine1RisingEdge
  - Spinnaker::Camera, [465](#)
- EventLine1RisingEdgeFrameID
  - Spinnaker::Camera, [465](#)
- EventLine1RisingEdgeTimestamp
  - Spinnaker::Camera, [465](#)
- EventLinkSpeedChange
  - Spinnaker::Camera, [466](#)
- EventLinkSpeedChangeFrameID
  - Spinnaker::Camera, [466](#)
- EventLinkSpeedChangeTimestamp
  - Spinnaker::Camera, [466](#)
- EventLinkTrigger0
  - Spinnaker::Camera, [466](#)
- EventLinkTrigger0FrameID
  - Spinnaker::Camera, [466](#)
- EventLinkTrigger0Timestamp
  - Spinnaker::Camera, [466](#)
- EventLinkTrigger1
  - Spinnaker::Camera, [466](#)
- EventLinkTrigger1FrameID
  - Spinnaker::Camera, [466](#)
- EventLinkTrigger1Timestamp
  - Spinnaker::Camera, [467](#)
- EventNotification
  - Spinnaker::Camera, [467](#)
- EventNotification\_Off
  - CameraDefs Class, [97](#)
- EventNotification\_On
  - CameraDefs Class, [97](#)
- EventNotificationEnums
  - CameraDefs Class, [96](#)
- EventPort Class, [223](#)
- EventProcessor
  - Spinnaker::Event, [637](#)
- EventSelector
  - Spinnaker::Camera, [467](#)
- EventSelector\_Error
  - CameraDefs Class, [97](#)
- EventSelector\_ExposureEnd
  - CameraDefs Class, [97](#)
- EventSelector\_SerialPortReceive
  - CameraDefs Class, [97](#)
- EventSelectorEnums
  - CameraDefs Class, [97](#)
- EventSequencerSetChange
  - Spinnaker::Camera, [467](#)
- EventSequencerSetChangeFrameID
  - Spinnaker::Camera, [467](#)
- EventSequencerSetChangeTimestamp
  - Spinnaker::Camera, [467](#)
- EventSerialData
  - Spinnaker::Camera, [467](#)
- EventSerialDataLength
  - Spinnaker::Camera, [467](#)
- EventSerialPortReceive
  - Spinnaker::Camera, [468](#)
- EventSerialPortReceiveTimestamp
  - Spinnaker::Camera, [468](#)
- EventSerialReceiveOverflow
  - Spinnaker::Camera, [468](#)
- EventStream0TransferBlockEnd
  - Spinnaker::Camera, [468](#)
- EventStream0TransferBlockEndFrameID
  - Spinnaker::Camera, [468](#)
- EventStream0TransferBlockEndTimestamp
  - Spinnaker::Camera, [468](#)
- EventStream0TransferBlockStart
  - Spinnaker::Camera, [468](#)
- EventStream0TransferBlockStartFrameID
  - Spinnaker::Camera, [469](#)
- EventStream0TransferBlockStartTimestamp
  - Spinnaker::Camera, [469](#)
- EventStream0TransferBlockTrigger
  - Spinnaker::Camera, [469](#)
- EventStream0TransferBlockTriggerFrameID
  - Spinnaker::Camera, [469](#)
- EventStream0TransferBlockTriggerTimestamp
  - Spinnaker::Camera, [469](#)
- EventStream0TransferBurstEnd
  - Spinnaker::Camera, [469](#)
- EventStream0TransferBurstEndFrameID
  - Spinnaker::Camera, [469](#)
- EventStream0TransferBurstEndTimestamp
  - Spinnaker::Camera, [470](#)
- EventStream0TransferBurstStart
  - Spinnaker::Camera, [470](#)
- EventStream0TransferBurstStartFrameID
  - Spinnaker::Camera, [470](#)
- EventStream0TransferBurstStartTimestamp
  - Spinnaker::Camera, [470](#)
- EventStream0TransferEnd
  - Spinnaker::Camera, [470](#)
- EventStream0TransferEndFrameID
  - Spinnaker::Camera, [470](#)
- EventStream0TransferEndTimestamp
  - Spinnaker::Camera, [470](#)
- EventStream0TransferOverflow
  - Spinnaker::Camera, [471](#)
- EventStream0TransferOverflowFrameID
  - Spinnaker::Camera, [471](#)
- EventStream0TransferOverflowTimestamp
  - Spinnaker::Camera, [471](#)
- EventStream0TransferPause
  - Spinnaker::Camera, [471](#)
- EventStream0TransferPauseFrameID
  - Spinnaker::Camera, [471](#)
- EventStream0TransferPauseTimestamp

- Spinnaker::Camera, [471](#)
- EventStream0TransferResume
  - Spinnaker::Camera, [471](#)
- EventStream0TransferResumeFrameID
  - Spinnaker::Camera, [472](#)
- EventStream0TransferResumeTimestamp
  - Spinnaker::Camera, [472](#)
- EventStream0TransferStart
  - Spinnaker::Camera, [472](#)
- EventStream0TransferStartFrameID
  - Spinnaker::Camera, [472](#)
- EventStream0TransferStartTimestamp
  - Spinnaker::Camera, [472](#)
- EventTest
  - Spinnaker::Camera, [472](#)
- EventTestTimestamp
  - Spinnaker::Camera, [472](#)
- EventTimer0End
  - Spinnaker::Camera, [473](#)
- EventTimer0EndFrameID
  - Spinnaker::Camera, [473](#)
- EventTimer0EndTimestamp
  - Spinnaker::Camera, [473](#)
- EventTimer0Start
  - Spinnaker::Camera, [473](#)
- EventTimer0StartFrameID
  - Spinnaker::Camera, [473](#)
- EventTimer0StartTimestamp
  - Spinnaker::Camera, [473](#)
- EventTimer1End
  - Spinnaker::Camera, [473](#)
- EventTimer1EndFrameID
  - Spinnaker::Camera, [473](#)
- EventTimer1EndTimestamp
  - Spinnaker::Camera, [474](#)
- EventTimer1Start
  - Spinnaker::Camera, [474](#)
- EventTimer1StartFrameID
  - Spinnaker::Camera, [474](#)
- EventTimer1StartTimestamp
  - Spinnaker::Camera, [474](#)
- EventType
  - Spinnaker Definitions, [163](#)
- Exception, [638](#)
  - Spinnaker::Exception, [640](#)
- Exception Class, [141](#)
- Execute
  - Spinnaker::GenApi::CommandNode, [597](#)
- Expert
  - Types Enums, [314](#)
- ExposureActiveMode
  - Spinnaker::Camera, [474](#)
- ExposureActiveMode\_AllPixels
  - CameraDefs Class, [97](#)
- ExposureActiveMode\_AnyPixels
  - CameraDefs Class, [97](#)
- ExposureActiveMode\_Line1
  - CameraDefs Class, [97](#)
- ExposureActiveModeEnums
  - CameraDefs Class, [97](#)
- ExposureAuto
  - Spinnaker::Camera, [474](#)
- ExposureAuto\_Continuous
  - CameraDefs Class, [97](#)
- ExposureAuto\_Off
  - CameraDefs Class, [97](#)
- ExposureAuto\_Once
  - CameraDefs Class, [97](#)
- ExposureAutoEnums
  - CameraDefs Class, [97](#)
- ExposureMode
  - Spinnaker::Camera, [474](#)
- ExposureMode\_Timed
  - CameraDefs Class, [98](#)
- ExposureMode\_TriggerWidth
  - CameraDefs Class, [98](#)
- ExposureModeEnums
  - CameraDefs Class, [97](#)
- ExposureTime
  - Spinnaker::Camera, [474](#)
- ExposureTimeMode
  - Spinnaker::Camera, [475](#)
- ExposureTimeMode\_Common
  - CameraDefs Class, [98](#)
- ExposureTimeMode\_Individual
  - CameraDefs Class, [98](#)
- ExposureTimeModeEnums
  - CameraDefs Class, [98](#)
- ExposureTimeSelector
  - Spinnaker::Camera, [475](#)
- ExposureTimeSelector\_Blue
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Common
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Cyan
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Green
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Infrared
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Magenta
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Red
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Stage1
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Stage2
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Ultraviolet
  - CameraDefs Class, [98](#)
- ExposureTimeSelector\_Yellow
  - CameraDefs Class, [98](#)
- ExposureTimeSelectorEnums
  - CameraDefs Class, [98](#)
- ExtractIndependentSubtree
  - INodeMapDyn Interface, [263](#)



- ExtractPolarization
  - Spinnaker::Image, [687](#)
  - Spinnaker::Image, [710](#)
- ExtractSubtree
  - Spinnaker::GenApi::CNodeMapFactory, [587](#)
- FLIR\_SPINNAKER\_VERSION\_BUILD
  - System.h, [1100](#)
- FLIR\_SPINNAKER\_VERSION\_MAJOR
  - System.h, [1100](#)
- FLIR\_SPINNAKER\_VERSION\_MINOR
  - System.h, [1100](#)
- FLIR\_SPINNAKER\_VERSION\_TYPE
  - System.h, [1100](#)
- FMT\_I64
  - Compatibility.h, [968](#)
- FROM\_FILE\_EXT
  - Spinnaker Definitions, [163](#)
- FULL\_RESOLUTION
  - Spinnaker Definitions, [166](#)
- FactoryReset
  - Spinnaker::Camera, [475](#)
- FileAccessBuffer
  - Spinnaker::Camera, [475](#)
- FileAccessLength
  - Spinnaker::Camera, [475](#)
- FileAccessOffset
  - Spinnaker::Camera, [475](#)
- FileOpenMode
  - Spinnaker::Camera, [475](#)
- FileOpenMode\_Read
  - CameraDefs Class, [99](#)
- FileOpenMode\_ReadWrite
  - CameraDefs Class, [99](#)
- FileOpenMode\_Write
  - CameraDefs Class, [99](#)
- FileOpenModeEnums
  - CameraDefs Class, [98](#)
- FileOperationExecute
  - Spinnaker::Camera, [476](#)
- FileOperationResult
  - Spinnaker::Camera, [476](#)
- FileOperationSelector
  - Spinnaker::Camera, [476](#)
- FileOperationSelector\_Close
  - CameraDefs Class, [99](#)
- FileOperationSelector\_Delete
  - CameraDefs Class, [99](#)
- FileOperationSelector\_Open
  - CameraDefs Class, [99](#)
- FileOperationSelector\_Read
  - CameraDefs Class, [99](#)
- FileOperationSelector\_Write
  - CameraDefs Class, [99](#)
- FileOperationSelectorEnums
  - CameraDefs Class, [99](#)
- FileOperationStatus
  - Spinnaker::Camera, [476](#)
- FileOperationStatus\_Failure
  - CameraDefs Class, [99](#)
- FileOperationStatus\_Overflow
  - CameraDefs Class, [99](#)
- FileOperationStatus\_Success
  - CameraDefs Class, [99](#)
- FileOperationStatusEnums
  - CameraDefs Class, [99](#)
- FileProtocolAdapter, [642](#)
  - Spinnaker::GenApi::FileProtocolAdapter, [643](#)
- FileSelector
  - Spinnaker::Camera, [476](#)
- FileSelector\_SerialPort0
  - CameraDefs Class, [99](#)
- FileSelector\_UserFile1
  - CameraDefs Class, [99](#)
- FileSelector\_UserSet0
  - CameraDefs Class, [99](#)
- FileSelector\_UserSet1
  - CameraDefs Class, [99](#)
- FileSelector\_UserSetDefault
  - CameraDefs Class, [99](#)
- FileSelectorEnums
  - CameraDefs Class, [99](#)
- FileSize
  - Spinnaker::Camera, [476](#)
- filebuf\_type
  - Spinnaker::GenApi::IDevFileStreamBase, [681](#)
  - Spinnaker::GenApi::ODevFileStreamBase, [786](#)
- Filestream Class, [224](#)
- find
  - Spinnaker::GenICam::gcstring, [655](#), [656](#)
- find\_first\_not\_of
  - Spinnaker::GenICam::gcstring, [656](#)
- find\_first\_of
  - Spinnaker::GenICam::gcstring, [656](#)
- fixedIncrement
  - Types Enums, [312](#)
- Flags
  - GVCP\_REQUEST\_HEADER, [664](#)
  - U3V\_COMMAND\_HEADER, [845](#)
- float32\_t
  - GCTypes Class, [229](#)
- float64\_t
  - GCTypes Class, [229](#)
- FloatNode, [645](#)
  - Spinnaker::GenApi::FloatNode, [647](#)
- FloatNode Class, [225](#)
  - CFloatRef, [225](#)
- FloatRegNode, [650](#)
  - Spinnaker::GenApi::FloatRegNode, [651](#)
- FloatRegNode Class, [226](#)
- fnAutomatic
  - Types Enums, [311](#)
- fnFixed
  - Types Enums, [311](#)
- fnScientific
  - Types Enums, [311](#)
- frameRate

- Spinnaker::Video::AVIOption, [379](#)
- Spinnaker::Video::H264Option, [665](#)
- Spinnaker::Video::MJPGOption, [770](#)
- FromString
  - IValue Class, [278](#)
  - Spinnaker::GenApi::EAccessModeClass, [619](#)
  - Spinnaker::GenApi::ECachingModeClass, [620](#)
  - Spinnaker::GenApi::EDisplayNotationClass, [621](#)
  - Spinnaker::GenApi::EEndianessClass, [622](#)
  - Spinnaker::GenApi::EGenApiSchemaVersion↔  
Class, [623](#)
  - Spinnaker::GenApi::EInputDirectionClass, [623](#)
  - Spinnaker::GenApi::ENameSpaceClass, [624](#)
  - Spinnaker::GenApi::ERepresentationClass, [632](#)
  - Spinnaker::GenApi::ESignClass, [633](#)
  - Spinnaker::GenApi::ESlopeClass, [634](#)
  - Spinnaker::GenApi::EStandardNameSpaceClass,  
[635](#)
  - Spinnaker::GenApi::EVisibilityClass, [638](#)
  - Spinnaker::GenApi::EYesNoClass, [642](#)
  - Spinnaker::GenApi::ValueNode, [848](#)
- Function\_NodeCallback
  - Spinnaker::GenApi::Function\_NodeCallback, [653](#)
- Function\_NodeCallback< Function >, [652](#)
- GC\_COUNTOF
  - GCUtilities.h, [1004](#)
- GC\_INT32\_MAX
  - GCTypes.h, [1000](#)
- GC\_INT32\_MIN
  - GCTypes.h, [1000](#)
- GC\_INT64\_MAX
  - GCTypes.h, [1000](#)
- GC\_INT64\_MIN
  - GCTypes.h, [1000](#)
- GC\_INT8\_MAX
  - GCTypes.h, [1000](#)
- GC\_INT8\_MIN
  - GCTypes.h, [1000](#)
- GC\_UINT32\_MAX
  - GCTypes.h, [1000](#)
- GC\_UINT64\_MAX
  - GCTypes.h, [1000](#)
- GC\_UINT8\_MAX
  - GCTypes.h, [1001](#)
- GCSTRING\_NPOS
  - GCString.h, [997](#)
- GCString Class, [227](#)
- GCString.h
  - GCSTRING\_NPOS, [997](#)
  - operator<<, [997](#)
  - operator>>, [997](#)
- GCSynch Class, [228](#)
- GCTypes Class, [229](#)
  - float32\_t, [229](#)
  - float64\_t, [229](#)
- GCTypes.h
  - \_\_STDC\_CONSTANT\_MACROS, [1000](#)
  - \_\_STDC\_LIMIT\_MACROS, [1000](#)
- GC\_INT32\_MAX, [1000](#)
- GC\_INT32\_MIN, [1000](#)
- GC\_INT64\_MAX, [1000](#)
- GC\_INT64\_MIN, [1000](#)
- GC\_INT8\_MAX, [1000](#)
- GC\_INT8\_MIN, [1000](#)
- GC\_UINT32\_MAX, [1000](#)
- GC\_UINT64\_MAX, [1000](#)
- GC\_UINT8\_MAX, [1001](#)
- GCUtilities Utility, [231](#)
  - DoesEnvironmentVariableExist, [232](#)
  - GetFiles, [232](#)
  - GetGenICamCLProtocolFolder, [232](#)
  - GetGenICamCacheFolder, [232](#)
  - GetGenICamLogConfig, [232](#)
  - GetModulePathFromFunction, [233](#)
  - GetValueOfEnvironmentVariable, [233](#)
  - INTEGRAL\_CAST2, [233](#)
  - INTEGRAL\_CAST, [233](#)
  - ReplaceEnvironmentVariables, [233](#)
  - SetGenICamCLProtocolFolder, [234](#)
  - SetGenICamCacheFolder, [234](#)
  - SetGenICamLogConfig, [234](#)
  - Tokenize, [234](#)
  - UrlDecode, [234](#)
  - UrlEncode, [234](#)
- GCUtilities.h
  - \_TO\_STRING, [1004](#)
  - \_\_ERR\_\_, [1004](#)
  - \_\_LINE\_STR\_\_, [1004](#)
  - \_\_LOCATION\_\_, [1004](#)
  - \_\_OUTPUT\_FORMATER\_\_, [1004](#)
  - \_\_TODO\_\_, [1004](#)
  - \_\_WARN\_\_, [1004](#)
  - EXPAND\_TO\_STRINGISE, [1004](#)
  - GC\_COUNTOF, [1004](#)
  - GENICAM\_DEPRECATED, [1004](#)
  - GENICAM\_UNUSED, [1004](#)
  - USE\_TEMP\_CACHE\_FILE, [1004](#)
- GENCP\_COMMAND\_HEADER\_SIZE
  - Spinnaker::GenApi, [372](#)
- GENCP\_EVENT\_BASIC\_SIZE
  - Spinnaker::GenApi, [372](#)
- GENCP\_EVENT\_CMD\_ID
  - Spinnaker::GenApi, [372](#)
- GENICAM\_DEPRECATED
  - GCUtilities.h, [1004](#)
- GENICAM\_ERR\_ACCESS
  - Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_BAD\_ALLOCATION
  - Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_DYNAMIC\_CAST
  - Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_GENERIC
  - Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_INVALID\_ARGUMENT
  - Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_LOGICAL



- Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_OUT\_OF\_RANGE
  - Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_PROPERTY
  - Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_RUN\_TIME
  - Spinnaker Definitions, [162](#)
- GENICAM\_ERR\_TIMEOUT
  - Spinnaker Definitions, [162](#)
- GENICAM\_UNUSED
  - GCUilities.h, [1004](#)
- GEV
  - Types Enums, [314](#)
- GREEN
  - Spinnaker Definitions, [167](#)
- GREY
  - Spinnaker Definitions, [167](#)
- GUIXMLLocation
  - Spinnaker::TransportLayerDevice, [831](#)
- GUIXMLLocation\_Device
  - TransportLayerDefs Class, [177](#)
- GUIXMLLocation\_Host
  - TransportLayerDefs Class, [177](#)
- GUIXMLLocationEnum
  - TransportLayerDefs Class, [177](#)
- GUIXMLPath
  - Spinnaker::TransportLayerDevice, [831](#)
- GVCP\_CHUNK\_TRAILER, [657](#)
  - ChunkId, [658](#)
  - ChunkLength, [658](#)
- GVCP\_EVENT\_ITEM\_BASIC, [659](#)
  - EventId, [659](#)
  - ReservedOrEventSize, [659](#)
- GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [659](#)
  - BlockId, [660](#)
  - BlockId64High, [660](#)
  - BlockId64Low, [660](#)
  - EventId, [660](#)
  - ReservedOrEventSize, [660](#)
  - StreamChannelId, [660](#)
  - TimestampHigh, [660](#)
  - TimestampLow, [660](#)
- GVCP\_EVENT\_ITEM, [658](#)
  - BlockId, [658](#)
  - EventId, [658](#)
  - ReservedOrEventSize, [658](#)
  - StreamChannelId, [658](#)
  - TimestampHigh, [658](#)
  - TimestampLow, [658](#)
- GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, [661](#)
  - Header, [662](#)
  - Items, [662](#)
- GVCP\_EVENT\_REQUEST, [660](#)
  - Header, [661](#)
  - Items, [661](#)
- GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID, [663](#)
  - Data, [663](#)
  - Event, [663](#)
  - Header, [663](#)
- GVCP\_EVENTDATA\_REQUEST, [662](#)
  - Data, [662](#)
  - Event, [662](#)
  - Header, [662](#)
- GVCP\_MESSAGE\_TAGS
  - Spinnaker::GenApi, [371](#)
- GVCP\_REQUEST\_HEADER, [663](#)
  - Command, [664](#)
  - Flags, [664](#)
  - Length, [664](#)
  - Magic, [664](#)
  - ReqId, [664](#)
- Gain
  - Spinnaker::Camera, [477](#)
- GainAuto
  - Spinnaker::Camera, [477](#)
- GainAuto\_Continuous
  - CameraDefs Class, [100](#)
- GainAuto\_Off
  - CameraDefs Class, [100](#)
- GainAuto\_Once
  - CameraDefs Class, [100](#)
- GainAutoBalance
  - Spinnaker::Camera, [477](#)
- GainAutoBalance\_Continuous
  - CameraDefs Class, [100](#)
- GainAutoBalance\_Off
  - CameraDefs Class, [100](#)
- GainAutoBalance\_Once
  - CameraDefs Class, [100](#)
- GainAutoBalanceEnums
  - CameraDefs Class, [99](#)
- GainAutoEnums
  - CameraDefs Class, [100](#)
- GainSelector
  - Spinnaker::Camera, [477](#)
- GainSelector\_All
  - CameraDefs Class, [100](#)
- GainSelectorEnums
  - CameraDefs Class, [100](#)
- Gamma
  - Spinnaker::Camera, [477](#)
- GammaEnable
  - Spinnaker::Camera, [477](#)
- gcstring, [654](#)
  - Spinnaker::GenICam::gcstring, [655](#)
- GenICamXMLLocation
  - Spinnaker::TransportLayerDevice, [829](#)
- GenICamXMLLocation\_Device
  - TransportLayerDefs Class, [177](#)
- GenICamXMLLocation\_Host
  - TransportLayerDefs Class, [177](#)
- GenICamXMLLocationEnum
  - TransportLayerDefs Class, [176](#)
- GenICamXMLPath
  - Spinnaker::TransportLayerDevice, [829](#)
- Get

- IRegister Interfaces, 272
  - Spinnaker::GenApi::RegisterNode, 802
- get
  - Spinnaker::BasePtr, 381
- GetAccessMode
  - Spinnaker::CameraBase, 516
  - Spinnaker::GenApi::CChunkPort, 541
  - Spinnaker::GenApi::CEventPort, 557
  - Spinnaker::GenApi::CPortImpl, 603
  - Spinnaker::GenApi::CRegisterPortImpl, 608
  - Spinnaker::GenApi::CTestPortStruct, 613
  - Spinnaker::GenApi::Node, 773
  - Spinnaker::GenApi::PortRecorder, 796
  - Spinnaker::ICameraBase, 669
- GetAddress
  - IRegister Interfaces, 272
  - Spinnaker::GenApi::RegisterNode, 802
- GetAlias
  - INode Interface, 255
  - Spinnaker::GenApi::Node, 773
- GetBitsPerPixel
  - Spinnaker::Image, 687
  - Spinnaker::Image, 710
- GetBlackLevel
  - Spinnaker::ChunkData, 568
  - Spinnaker::IChunkData, 676
- getBufSize
  - Spinnaker::GenApi::FileProtocolAdapter, 644
- GetBufferSize
  - Spinnaker::Image, 687
  - Spinnaker::Image, 710
- GetBuildDate
  - Spinnaker::Exception, 640
- GetBuildTime
  - Spinnaker::Exception, 640
- GetByIndex
  - Spinnaker::CameraList, 524
  - Spinnaker::ICameraList, 674
  - Spinnaker::IInterfaceList, 701
  - Spinnaker::InterfaceList, 747
- GetBySerial
  - Spinnaker::CameraList, 524
  - Spinnaker::ICameraList, 674
- GetCRC
  - Spinnaker::ChunkData, 568
  - Spinnaker::IChunkData, 676
- GetCachingMode
  - INode Interface, 255
  - Spinnaker::GenApi::Node, 774
- GetCallbackType
  - Spinnaker::GenApi::CNodeCallback, 581
- GetCameras
  - Spinnaker::IInterface, 697
  - Spinnaker::ISystem, 754
  - Spinnaker::Interface, 741
  - Spinnaker::System, 817
- GetCastAlias
  - INode Interface, 255
- Spinnaker::GenApi::Node, 774
- GetCategoryName
  - Spinnaker::LoggingEventData, 764
- GetChannelStatus
  - Spinnaker::ImageStatistics, 695
  - Spinnaker::ImageStatistics, 730
- GetChildren
  - INode Interface, 255
  - Spinnaker::GenApi::Node, 774
- GetChunkData
  - Spinnaker::Image, 687
  - Spinnaker::Image, 710
- GetChunkIDLength
  - Spinnaker::GenApi::CChunkPort, 541
- GetChunkID
  - Spinnaker::GenApi::PortNode, 792
- GetChunkLayoutId
  - Spinnaker::Image, 687
  - Spinnaker::Image, 711
- GetColorProcessing
  - Spinnaker::Image, 688
  - Spinnaker::Image, 711
- GetCookie
  - IPortRecorder Interface, 270
  - Spinnaker::GenApi::CPortWriteList, 606
- GetCounterValue
  - Spinnaker::ChunkData, 568
  - Spinnaker::IChunkData, 676
- GetCurrentEntry
  - IEnumeration Interface, 245
  - Spinnaker::GenApi::CEnumerationTRef, 544
  - Spinnaker::GenApi::EnumNode, 630
- GetData
  - Spinnaker::Image, 688
  - Spinnaker::Image, 711
- GetDefaultColorProcessing
  - Spinnaker::Image, 711
- GetDescription
  - INode Interface, 256
  - Spinnaker::GenApi::Node, 774
- GetDeviceEventId
  - Spinnaker::DeviceEvent, 617
  - Spinnaker::IDeviceEvent, 684
- GetDeviceEventName
  - Spinnaker::DeviceEvent, 617
  - Spinnaker::IDeviceEvent, 684
- GetDeviceName
  - INodeMap Interface, 261
  - Spinnaker::GenApi::Node, 774
  - Spinnaker::GenApi::NodeMap, 781
- GetDeviceVersion
  - IDeviceInfo Interface, 242
  - Spinnaker::GenApi::NodeMap, 781
- GetDisplayName
  - INode Interface, 256
  - Spinnaker::GenApi::Node, 774
- GetDisplayNotation
  - IFloat Interface, 250

- Spinnaker::GenApi::FloatNode, 647
- GetDisplayPrecision
  - IFloat Interface, 250
  - Spinnaker::GenApi::FloatNode, 647
- GetDocuURL
  - INode Interface, 256
  - Spinnaker::GenApi::Node, 774
- GetEncoderValue
  - Spinnaker::ChunkData, 569
  - Spinnaker::IChunkData, 676
- GetEntries
  - IEnumeration Interface, 245
  - Spinnaker::GenApi::EnumNode, 630
- GetEntry
  - IEnumeration Interface, 245
  - IEnumerationT Interface, 247
  - Spinnaker::GenApi::CEnumerationTRef, 544
  - Spinnaker::GenApi::EnumNode, 630
- GetEntryByName
  - IEnumeration Interface, 246
  - Spinnaker::GenApi::EnumNode, 630
- GetEnumAlias
  - Spinnaker::GenApi::CFloatPtr, 561
  - Spinnaker::GenApi::FloatNode, 647
- GetError
  - Spinnaker::Exception, 640
- GetErrorMessage
  - SpinUpdate.h, 1096
  - Spinnaker::Exception, 640
- GetEventIDLength
  - Spinnaker::GenApi::CEventPort, 557
- GetEventID
  - INode Interface, 256
  - Spinnaker::GenApi::Node, 774
- GetEventPayloadData
  - Spinnaker::Event, 636
- GetEventPayloadDataSize
  - Spinnaker::Event, 636
- GetEventType
  - Spinnaker::Event, 636
- GetExposureEndLineStatusAll
  - Spinnaker::ChunkData, 569
  - Spinnaker::IChunkData, 676
- GetExposureTime
  - Spinnaker::ChunkData, 569
  - Spinnaker::IChunkData, 676
- GetFeatureBagHandle
  - Spinnaker::GenApi::CFeatureBag, 559
- GetFeatures
  - Spinnaker::GenApi::CategoryNode, 528
- GetFileName
  - Spinnaker::Exception, 640
- GetFiles
  - GCUtilities Utility, 232
- GetFloatAlias
  - Spinnaker::GenApi::IntegerNode, 737
- GetFrameID
  - Spinnaker::ChunkData, 569
- Spinnaker::IChunkData, 677
- Spinnaker::Image, 688
- Spinnaker::Image, 712
- GetFullErrorMessage
  - Spinnaker::Exception, 640
- GetFunctionName
  - Spinnaker::Exception, 640
- GetGain
  - Spinnaker::ChunkData, 569
  - Spinnaker::IChunkData, 677
- GetGenApiVersion
  - IDeviceInfo Interface, 242
  - Spinnaker::GenApi::NodeMap, 781
- GetGenICamCLProtocolFolder
  - GCUtilities Utility, 232
- GetGenICamCacheFolder
  - GCUtilities Utility, 232
- GetGenICamLogConfig
  - GCUtilities Utility, 232
- GetGuiXml
  - Spinnaker::CameraBase, 516
  - Spinnaker::ICameraBase, 670
- GetHeatMapColorGradient
  - Spinnaker::Image, 712
- GetHeatMapRange
  - Spinnaker::Image, 712
- GetHeight
  - Spinnaker::ChunkData, 569
  - Spinnaker::IChunkData, 677
  - Spinnaker::Image, 688
  - Spinnaker::Image, 713
- GetHistogram
  - Spinnaker::ImageStatistics, 695
  - Spinnaker::ImageStatistics, 731
- GetID
  - Spinnaker::Image, 688
  - Spinnaker::Image, 713
- GetImage
  - Spinnaker::ChunkData, 570
  - Spinnaker::IChunkData, 677
- GetImageSize
  - Spinnaker::Image, 688
  - Spinnaker::Image, 713
- GetImageStatus
  - Spinnaker::Image, 688
  - Spinnaker::Image, 713
- GetImageStatusDescription
  - Spinnaker::Image, 713
- GetInc
  - IFloat Interface, 250
  - Spinnaker::GenApi::FloatNode, 647
  - Spinnaker::GenApi::IntegerNode, 737
- GetIncMode
  - IFloat Interface, 250
  - Spinnaker::GenApi::FloatNode, 648
  - Spinnaker::GenApi::IntegerNode, 737
- GetInferenceConfidence
  - Spinnaker::ChunkData, 570

- Spinnaker::IChunkData, 677
- GetInferenceResult
  - Spinnaker::ChunkData, 570
  - Spinnaker::IChunkData, 677
- GetInstance
  - Spinnaker::System, 817
- GetIntAlias
  - Spinnaker::GenApi::CFloatPtr, 561
  - Spinnaker::GenApi::FloatNode, 648
- GetIntValue
  - IEnumeration Interface, 246
  - Spinnaker::GenApi::EnumNode, 630
- GetInterfaceName
  - Pointer Class, 291
- GetInterfaces
  - Spinnaker::ISystem, 754
  - Spinnaker::System, 818
- GetLength
  - IRegister Interfaces, 273
  - Spinnaker::GenApi::RegisterNode, 802
- GetLibraryVersion
  - Spinnaker::ISystem, 754
  - Spinnaker::System, 818
- GetLineNumber
  - Spinnaker::Exception, 641
- GetLinePitch
  - Spinnaker::ChunkData, 570
  - Spinnaker::IChunkData, 677
- GetLineStatusAll
  - Spinnaker::ChunkData, 570
  - Spinnaker::IChunkData, 677
- GetListOfValidValues
  - IFloat Interface, 250
  - Spinnaker::GenApi::FloatNode, 648
  - Spinnaker::GenApi::IntegerNode, 737
- GetLock
  - INodeMap Interface, 261
  - Spinnaker::GenApi::NodeMap, 781
  - Spinnaker::GenICam::LockableObject, 761
- GetLogMessage
  - Spinnaker::LoggingEventData, 764
- GetLoggingEventPriorityLevel
  - Spinnaker::ISystem, 755
  - Spinnaker::System, 818
- GetMax
  - IFloat Interface, 250
  - Spinnaker::GenApi::FloatNode, 648
  - Spinnaker::GenApi::IntegerNode, 737
- GetMaxLength
  - IString Class, 277
  - Spinnaker::GenApi::StringNode, 811
- GetMean
  - Spinnaker::IImageStatistics, 695
  - Spinnaker::ImageStatistics, 731
- GetMin
  - IFloat Interface, 250
  - Spinnaker::GenApi::FloatNode, 648
  - Spinnaker::GenApi::IntegerNode, 738
- GetModelName
  - Spinnaker::GenApi::NodeMap, 781
- GetModulePathFromFunction
  - GCUilities Utility, 233
- GetNDC
  - Spinnaker::LoggingEventData, 764
- GetName
  - Spinnaker::GenApi::Node, 774
- GetNameSpace
  - INode Interface, 256
  - Spinnaker::GenApi::Node, 775
- GetNextImage
  - Spinnaker::CameraBase, 517
  - Spinnaker::ICameraBase, 670
- GetNode
  - INodeMap Interface, 261
  - Spinnaker::GenApi::CNodeCallback, 581
  - Spinnaker::GenApi::NodeMap, 781
  - Spinnaker::GenApi::ValueNode, 849
- GetNodeHandle
  - Spinnaker::GenApi::Node, 775
- GetNodeMap
  - INode Interface, 256
  - Spinnaker::CameraBase, 517
  - Spinnaker::GenApi::Node, 775
  - Spinnaker::ICameraBase, 670
- GetNodeMapHandle
  - Spinnaker::GenApi::NodeMap, 781
- GetNodeStatistics
  - Spinnaker::GenApi::CNodeMapFactory, 587
- GetNodes
  - Spinnaker::GenApi::NodeMap, 781
- GetNumChannels
  - Spinnaker::IImage, 688
  - Spinnaker::Image, 714
- GetNumDataStreams
  - Spinnaker::CameraBase, 517
  - Spinnaker::ICameraBase, 670
- GetNumImagesInUse
  - Spinnaker::CameraBase, 518
  - Spinnaker::ICameraBase, 670
- GetNumNodes
  - INodeMap Interface, 261
  - Spinnaker::GenApi::NodeMap, 781
- GetNumPixelValues
  - Spinnaker::IImageStatistics, 695
  - Spinnaker::ImageStatistics, 731
- GetNumReads
  - Spinnaker::GenApi::CTestPortStruct, 613
- GetNumWrites
  - Spinnaker::GenApi::CTestPortStruct, 613
- GetNumericValue
  - IEnumEntry Interface, 244
  - Spinnaker::GenApi::EnumEntryNode, 627
- GetOffsetX
  - Spinnaker::ChunkData, 570
  - Spinnaker::IChunkData, 677
- GetOffsetY

- Spinnaker::ChunkData, 571
- Spinnaker::IChunkData, 677
- GetParents
  - INode Interface, 256
  - Spinnaker::GenApi::Node, 775
- GetPartSelector
  - Spinnaker::ChunkData, 571
  - Spinnaker::IChunkData, 678
- GetPayloadType
  - Spinnaker::Image, 688
  - Spinnaker::Image, 714
- GetPixelDynamicRangeMax
  - Spinnaker::ChunkData, 571
  - Spinnaker::IChunkData, 678
- GetPixelDynamicRangeMin
  - Spinnaker::ChunkData, 571
  - Spinnaker::IChunkData, 678
- GetPixelFormat
  - Spinnaker::Image, 688
  - Spinnaker::Image, 714
- GetPixelFormatIntType
  - Spinnaker::Image, 689
  - Spinnaker::Image, 714
- GetPixelFormatName
  - Spinnaker::Image, 689
  - Spinnaker::Image, 715
- GetPixelValueRange
  - Spinnaker::ImageStatistics, 695
  - Spinnaker::ImageStatistics, 731
- GetPolarizationAlgorithm
  - Spinnaker::Image, 689
  - Spinnaker::Image, 715
- GetPolarizationValues
  - Spinnaker::Image, 689
  - Spinnaker::Image, 715
- GetPollingTime
  - INode Interface, 256
  - Spinnaker::GenApi::Node, 775
- GetPortHandle
  - Spinnaker::GenApi::PortNode, 792
- GetPortReplayHandle
  - Spinnaker::GenApi::PortReplay, 798
- GetPortWriteListHandle
  - Spinnaker::GenApi::CPortWriteList, 606
- GetPrincipalInterfaceType
  - INode Interface, 256
  - Spinnaker::GenApi::CChunkPort, 541
  - Spinnaker::GenApi::CEventPort, 557
  - Spinnaker::GenApi::CTestPortStruct, 613
  - Spinnaker::GenApi::Node, 775
- GetPriority
  - Spinnaker::LoggingEventData, 765
- GetPriorityName
  - Spinnaker::LoggingEventData, 765
- GetPrivateData
  - Spinnaker::Image, 689
  - Spinnaker::Image, 715
- GetProductGuid
  - IDeviceInfo Interface, 243
  - Spinnaker::GenApi::NodeMap, 782
- GetProperty
  - INode Interface, 257
  - Spinnaker::GenApi::Node, 775
- GetPropertyNames
  - INode Interface, 257
  - Spinnaker::GenApi::Node, 775
- GetRange
  - Spinnaker::ImageStatistics, 695
  - Spinnaker::ImageStatistics, 732
- GetRepresentation
  - IFloat Interface, 250
  - Spinnaker::GenApi::FloatNode, 648
  - Spinnaker::GenApi::IntegerNode, 738
- GetScan3dAxisMax
  - Spinnaker::ChunkData, 571
  - Spinnaker::IChunkData, 678
- GetScan3dAxisMin
  - Spinnaker::ChunkData, 571
  - Spinnaker::IChunkData, 678
- GetScan3dCoordinateOffset
  - Spinnaker::ChunkData, 572
  - Spinnaker::IChunkData, 678
- GetScan3dCoordinateReferenceValue
  - Spinnaker::ChunkData, 572
  - Spinnaker::IChunkData, 678
- GetScan3dCoordinateScale
  - Spinnaker::ChunkData, 572
  - Spinnaker::IChunkData, 678
- GetScan3dInvalidDataValue
  - Spinnaker::ChunkData, 572
  - Spinnaker::IChunkData, 678
- GetScan3dTransformValue
  - Spinnaker::ChunkData, 572
  - Spinnaker::IChunkData, 678
- GetScanLineSelector
  - Spinnaker::ChunkData, 572
  - Spinnaker::IChunkData, 679
- GetSchemaVersion
  - IDeviceInfo Interface, 243
  - Spinnaker::GenApi::NodeMap, 782
- GetSelectedFeatures
  - ISelector Interface, 274
  - Spinnaker::GenApi::Node, 775
- GetSelectingFeatures
  - ISelector Interface, 274
  - Spinnaker::GenApi::Node, 776
- GetSelectorList
  - ISelectorDigit Interface, 275
  - Spinnaker::GenApi::CSelectorSet, 611
- GetSequencerSetActive
  - Spinnaker::ChunkData, 573
  - Spinnaker::IChunkData, 679
- GetSerialDataLength
  - Spinnaker::ChunkData, 573
  - Spinnaker::IChunkData, 679
- GetSize

- Spinnaker::CameraList, [524](#)
- Spinnaker::ICameraList, [674](#)
- Spinnaker::IInterfaceList, [701](#)
- Spinnaker::InterfaceList, [747](#)
- GetStandardNameSpace
  - IDeviceInfo Interface, [243](#)
  - Spinnaker::GenApi::NodeMap, [782](#)
- GetStatistics
  - Spinnaker::IImageStatistics, [695](#)
  - Spinnaker::ImageStatistics, [732](#)
- GetStreamChannelID
  - Spinnaker::ChunkData, [573](#)
  - Spinnaker::IChunkData, [679](#)
- GetStride
  - Spinnaker::IImage, [689](#)
  - Spinnaker::Image, [716](#)
- GetSupportedSchemaVersions
  - INodeMapDyn Interface, [263](#)
  - Spinnaker::GenApi::CNodeMapFactory, [587](#)
  - Spinnaker::GenApi::NodeMap, [782](#)
- GetSwapEndianness
  - IPortConstruct Interface, [269](#)
  - Spinnaker::GenApi::CChunkPort, [541](#)
  - Spinnaker::GenApi::CEventPort, [557](#)
  - Spinnaker::GenApi::CPortImpl, [603](#)
  - Spinnaker::GenApi::PortNode, [792](#)
- GetSymbolic
  - IEnumEntry Interface, [244](#)
  - Spinnaker::GenApi::EnumEntryNode, [627](#)
- GetSymbolics
  - Spinnaker::GenApi::EnumNode, [631](#)
- GetTLDeviceNodeMap
  - Spinnaker::CameraBase, [518](#)
  - Spinnaker::ICameraBase, [670](#)
- GetTLNodeMap
  - Spinnaker::IInterface, [697](#)
  - Spinnaker::Interface, [741](#)
- GetTLPayloadType
  - Spinnaker::IImage, [689](#)
  - Spinnaker::Image, [716](#)
- GetTLPixelFormat
  - Spinnaker::IImage, [689](#)
  - Spinnaker::Image, [716](#)
- GetTLPixelFormatNamespace
  - Spinnaker::IImage, [689](#)
  - Spinnaker::Image, [717](#)
- GetTLStreamNodeMap
  - Spinnaker::CameraBase, [518](#)
  - Spinnaker::ICameraBase, [670](#)
- GetThreadName
  - Spinnaker::LoggingEventData, [765](#)
- GetTimeStamp
  - Spinnaker::IImage, [689](#)
  - Spinnaker::Image, [716](#)
- GetTimerValue
  - Spinnaker::ChunkData, [573](#)
  - Spinnaker::IChunkData, [679](#)
- GetTimestamp
  - Spinnaker::ChunkData, [573](#)
  - Spinnaker::IChunkData, [679](#)
- GetTimestampLatchValue
  - Spinnaker::ChunkData, [573](#)
  - Spinnaker::IChunkData, [679](#)
- GetToolTip
  - IDeviceInfo Interface, [243](#)
  - Spinnaker::GenApi::Node, [776](#)
  - Spinnaker::GenApi::NodeMap, [782](#)
- GetTransferBlockID
  - Spinnaker::ChunkData, [574](#)
  - Spinnaker::IChunkData, [679](#)
- GetTransferQueueCurrentBlockCount
  - Spinnaker::ChunkData, [574](#)
  - Spinnaker::IChunkData, [679](#)
- GetUniqueID
  - Spinnaker::CameraBase, [518](#)
  - Spinnaker::ICameraBase, [670](#)
- GetUnit
  - IFloat Interface, [250](#)
  - Spinnaker::GenApi::FloatNode, [648](#)
  - Spinnaker::GenApi::IntegerNode, [738](#)
- GetValidPayloadSize
  - Spinnaker::IImage, [690](#)
  - Spinnaker::Image, [717](#)
- GetValue
  - IBoolean Interface, [235](#)
  - Spinnaker::GenApi::BooleanNode, [385](#)
  - Spinnaker::GenApi::CEnumerationTRef, [544](#)
  - Spinnaker::GenApi::Counter, [598](#)
  - Spinnaker::GenApi::EnumEntryNode, [627](#)
  - Spinnaker::GenApi::FloatNode, [648](#)
  - Spinnaker::GenApi::IntegerNode, [738](#)
  - Spinnaker::GenApi::StringNode, [811](#)
- GetValueOfEnvironmentVariable
  - GCUtilities Utility, [233](#)
- GetVendorName
  - IDeviceInfo Interface, [243](#)
  - Spinnaker::GenApi::NodeMap, [782](#)
- GetVersionGuid
  - IDeviceInfo Interface, [243](#)
  - Spinnaker::GenApi::NodeMap, [783](#)
- GetVisibility
  - INode Interface, [257](#)
  - Spinnaker::GenApi::Node, [776](#)
- GetWidth
  - Spinnaker::ChunkData, [574](#)
  - Spinnaker::IChunkData, [679](#)
  - Spinnaker::IImage, [690](#)
  - Spinnaker::Image, [717](#)
- GetXOffset
  - Spinnaker::IImage, [690](#)
  - Spinnaker::Image, [718](#)
- GetXPadding
  - Spinnaker::IImage, [690](#)
  - Spinnaker::Image, [718](#)
- GetYOffset



- Spinnaker::Image, [690](#)
- Spinnaker::Image, [718](#)
- GetYPadding
  - Spinnaker::Image, [690](#)
  - Spinnaker::Image, [718](#)
- getline
  - Spinnaker::GenICam, [374](#)
- GevActionDeviceKey
  - Spinnaker::TransportLayerInterface, [835](#)
- GevActionGroupKey
  - Spinnaker::TransportLayerInterface, [835](#)
- GevActionGroupMask
  - Spinnaker::TransportLayerInterface, [836](#)
- GevActionTime
  - Spinnaker::TransportLayerInterface, [836](#)
- GevActiveLinkCount
  - Spinnaker::Camera, [477](#)
- GevCCP\_ControlAccess
  - CameraDefs Class, [100](#)
- GevCCP\_EnumEntry\_GevCCP\_ControlAccess
  - TransportLayerDefs Class, [177](#)
- GevCCP\_EnumEntry\_GevCCP\_ExclusiveAccess
  - TransportLayerDefs Class, [177](#)
- GevCCP\_EnumEntry\_GevCCP\_OpenAccess
  - TransportLayerDefs Class, [177](#)
- GevCCP\_ExclusiveAccess
  - CameraDefs Class, [100](#)
- GevCCP\_OpenAccess
  - CameraDefs Class, [100](#)
- GevCCPEnum
  - TransportLayerDefs Class, [177](#)
- GevCCPEnums
  - CameraDefs Class, [100](#)
- GevCCP
  - Spinnaker::Camera, [478](#)
  - Spinnaker::TransportLayerDevice, [829](#)
- GevCurrentDefaultGateway
  - Spinnaker::Camera, [478](#)
- GevCurrentIPAddress
  - Spinnaker::Camera, [478](#)
- GevCurrentIPConfigurationDHCP
  - Spinnaker::Camera, [478](#)
- GevCurrentIPConfigurationLLA
  - Spinnaker::Camera, [478](#)
- GevCurrentIPConfigurationPersistentIP
  - Spinnaker::Camera, [478](#)
- GevCurrentPhysicalLinkConfiguration
  - Spinnaker::Camera, [478](#)
- GevCurrentPhysicalLinkConfiguration\_DynamicLAG
  - CameraDefs Class, [101](#)
- GevCurrentPhysicalLinkConfiguration\_MultiLink
  - CameraDefs Class, [101](#)
- GevCurrentPhysicalLinkConfiguration\_SingleLink
  - CameraDefs Class, [101](#)
- GevCurrentPhysicalLinkConfiguration\_StaticLAG
  - CameraDefs Class, [101](#)
- GevCurrentPhysicalLinkConfigurationEnums
  - CameraDefs Class, [100](#)
- GevCurrentSubnetMask
  - Spinnaker::Camera, [478](#)
- GevDeviceDiscoverMaximumPacketSize
  - Spinnaker::TransportLayerDevice, [829](#)
- GevDeviceGateway
  - Spinnaker::TransportLayerDevice, [829](#)
- GevDeviceIPAddress
  - Spinnaker::TransportLayerDevice, [830](#)
  - Spinnaker::TransportLayerInterface, [836](#)
- GevDevicesWrongSubnet
  - Spinnaker::TransportLayerDevice, [830](#)
- GevDeviceMACAddress
  - Spinnaker::TransportLayerDevice, [830](#)
  - Spinnaker::TransportLayerInterface, [836](#)
- GevDeviceMaximumPacketSize
  - Spinnaker::TransportLayerDevice, [830](#)
- GevDeviceMaximumRetryCount
  - Spinnaker::TransportLayerDevice, [830](#)
- GevDeviceModelsBigEndian
  - Spinnaker::TransportLayerDevice, [830](#)
- GevDevicePort
  - Spinnaker::TransportLayerDevice, [830](#)
- GevDeviceReadAndWriteTimeout
  - Spinnaker::TransportLayerDevice, [831](#)
- GevDeviceSubnetMask
  - Spinnaker::TransportLayerDevice, [831](#)
  - Spinnaker::TransportLayerInterface, [836](#)
- GevDiscoveryAckDelay
  - Spinnaker::Camera, [479](#)
- GevFailedPacketCount
  - Spinnaker::TransportLayerStream, [841](#)
- GevFirstURL
  - Spinnaker::Camera, [479](#)
- GevGVCPExtendedStatusCodes
  - Spinnaker::Camera, [479](#)
- GevGVCPExtendedStatusCodesSelector
  - Spinnaker::Camera, [479](#)
- GevGVCPExtendedStatusCodesSelector\_Version1\_1
  - CameraDefs Class, [101](#)
- GevGVCPExtendedStatusCodesSelector\_Version2\_0
  - CameraDefs Class, [101](#)
- GevGVCPExtendedStatusCodesSelectorEnums
  - CameraDefs Class, [101](#)
- GevGVCPHeartbeatDisable
  - Spinnaker::Camera, [479](#)
- GevGVCPPendingAck
  - Spinnaker::Camera, [479](#)
- GevGVCPPendingTimeout
  - Spinnaker::Camera, [479](#)
- GevGVSPExtendedIDMode
  - Spinnaker::Camera, [480](#)
- GevGVSPExtendedIDMode\_Off
  - CameraDefs Class, [101](#)
- GevGVSPExtendedIDMode\_On
  - CameraDefs Class, [101](#)
- GevGVSPExtendedIDModeEnums
  - CameraDefs Class, [101](#)
- GevHeartbeatTimeout

- Spinnaker::Camera, [480](#)
- GevIEEE1588
  - Spinnaker::Camera, [480](#)
- GevIEEE1588ClockAccuracy
  - Spinnaker::Camera, [480](#)
- GevIEEE1588ClockAccuracy\_Unknown
  - CameraDefs Class, [101](#)
- GevIEEE1588ClockAccuracyEnums
  - CameraDefs Class, [101](#)
- GevIEEE1588Mode
  - Spinnaker::Camera, [480](#)
- GevIEEE1588Mode\_Auto
  - CameraDefs Class, [102](#)
- GevIEEE1588Mode\_SlaveOnly
  - CameraDefs Class, [102](#)
- GevIEEE1588ModeEnums
  - CameraDefs Class, [101](#)
- GevIEEE1588Status
  - Spinnaker::Camera, [480](#)
- GevIEEE1588Status\_Disabled
  - CameraDefs Class, [102](#)
- GevIEEE1588Status\_Faulty
  - CameraDefs Class, [102](#)
- GevIEEE1588Status\_Initializing
  - CameraDefs Class, [102](#)
- GevIEEE1588Status\_Listening
  - CameraDefs Class, [102](#)
- GevIEEE1588Status\_Master
  - CameraDefs Class, [102](#)
- GevIEEE1588Status\_Passive
  - CameraDefs Class, [102](#)
- GevIEEE1588Status\_PreMaster
  - CameraDefs Class, [102](#)
- GevIEEE1588Status\_Slave
  - CameraDefs Class, [102](#)
- GevIEEE1588Status\_Uncalibrated
  - CameraDefs Class, [102](#)
- GevIEEE1588StatusEnums
  - CameraDefs Class, [102](#)
- GevIPConfigurationStatus
  - Spinnaker::Camera, [481](#)
- GevIPConfigurationStatus\_DHCP
  - CameraDefs Class, [102](#)
- GevIPConfigurationStatus\_ForceIP
  - CameraDefs Class, [102](#)
- GevIPConfigurationStatus\_LLA
  - CameraDefs Class, [102](#)
- GevIPConfigurationStatus\_None
  - CameraDefs Class, [102](#)
- GevIPConfigurationStatus\_PersistentIP
  - CameraDefs Class, [102](#)
- GevIPConfigurationStatusEnums
  - CameraDefs Class, [102](#)
- GevInterfaceGateway
  - Spinnaker::TransportLayerInterface, [836](#)
- GevInterfaceIPAddress
  - Spinnaker::TransportLayerInterface, [836](#)
- GevInterfaceMACAddress
  - Spinnaker::TransportLayerInterface, [837](#)
- GevInterfaceSelector
  - Spinnaker::Camera, [480](#)
- GevInterfaceSubnetMask
  - Spinnaker::TransportLayerInterface, [837](#)
- GevMACAddress
  - Spinnaker::Camera, [481](#)
- GevMCDA
  - Spinnaker::Camera, [481](#)
- GevMCPHostPort
  - Spinnaker::Camera, [481](#)
- GevMCRC
  - Spinnaker::Camera, [481](#)
- GevMCSP
  - Spinnaker::Camera, [481](#)
- GevMCTT
  - Spinnaker::Camera, [481](#)
- GevMaximumNumberResendBuffers
  - Spinnaker::TransportLayerStream, [841](#)
- GevMaximumNumberResendRequests
  - Spinnaker::TransportLayerStream, [841](#)
- GevNumberOfInterfaces
  - Spinnaker::Camera, [481](#)
- GevPAUSEFrameReception
  - Spinnaker::Camera, [482](#)
- GevPAUSEFrameTransmission
  - Spinnaker::Camera, [482](#)
- GevPacketResendMode
  - Spinnaker::TransportLayerStream, [841](#)
- GevPacketResendTimeout
  - Spinnaker::TransportLayerStream, [841](#)
- GevPersistentDefaultGateway
  - Spinnaker::Camera, [482](#)
- GevPersistentIPAddress
  - Spinnaker::Camera, [482](#)
- GevPersistentSubnetMask
  - Spinnaker::Camera, [482](#)
- GevPhysicalLinkConfiguration
  - Spinnaker::Camera, [482](#)
- GevPhysicalLinkConfiguration\_DynamicLAG
  - CameraDefs Class, [102](#)
- GevPhysicalLinkConfiguration\_MultiLink
  - CameraDefs Class, [102](#)
- GevPhysicalLinkConfiguration\_SingleLink
  - CameraDefs Class, [102](#)
- GevPhysicalLinkConfiguration\_StaticLAG
  - CameraDefs Class, [102](#)
- GevPhysicalLinkConfigurationEnums
  - CameraDefs Class, [102](#)
- GevPrimaryApplicationIPAddress
  - Spinnaker::Camera, [482](#)
- GevPrimaryApplicationSocket
  - Spinnaker::Camera, [482](#)
- GevPrimaryApplicationSwitchoverKey
  - Spinnaker::Camera, [483](#)
- GevResendPacketCount
  - Spinnaker::TransportLayerStream, [841](#)
- GevResendRequestCount



- Spinnaker::TransportLayerStream, [842](#)
- GevSCCFGAllInTransmission
  - Spinnaker::Camera, [483](#)
- GevSCCFGExtendedChunkData
  - Spinnaker::Camera, [483](#)
- GevSCCFGPacketResendDestination
  - Spinnaker::Camera, [483](#)
- GevSCCFGUnconditionalStreaming
  - Spinnaker::Camera, [483](#)
- GevSCDA
  - Spinnaker::Camera, [483](#)
- GevSCPDDirection
  - Spinnaker::Camera, [484](#)
- GevSCPHostPort
  - Spinnaker::Camera, [484](#)
- GevSCPIInterfaceIndex
  - Spinnaker::Camera, [484](#)
- GevSCPSBigEndian
  - Spinnaker::Camera, [484](#)
- GevSCPSDoNotFragment
  - Spinnaker::Camera, [484](#)
- GevSCPSFireTestPacket
  - Spinnaker::Camera, [484](#)
- GevSCPSPacketSize
  - Spinnaker::Camera, [484](#)
- GevSCPD
  - Spinnaker::Camera, [483](#)
- GevSCSP
  - Spinnaker::Camera, [485](#)
- GevSCZoneConfigurationLock
  - Spinnaker::Camera, [485](#)
- GevSCZoneCount
  - Spinnaker::Camera, [485](#)
- GevSCZoneDirectionAll
  - Spinnaker::Camera, [485](#)
- GevSecondURL
  - Spinnaker::Camera, [485](#)
- GevStreamChannelSelector
  - Spinnaker::Camera, [485](#)
- GevSupportedOption
  - Spinnaker::Camera, [485](#)
- GevSupportedOptionSelector
  - Spinnaker::Camera, [486](#)
- GevSupportedOptionSelector\_Action
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_CCPApplicationSocket
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_CommandsConcatenation
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_DiscoveryAckDelay
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_DiscoveryAckDelay↔
  - Writable
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_Event
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_EventData
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_ExtendedStatusCodes
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_HeartbeatDisable
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_IPConfigurationDHCP
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_IPConfigurationLLA
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_IPConfiguration↔
  - PersistentIP
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_LinkSpeed
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_ManifestTable
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_MessageChannel↔
  - SourceSocket
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_PacketResend
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_PendingAck
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_SerialNumber
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_StreamChannel↔
  - SourceSocket
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_TestData
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_UserDefinedName
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelector\_WriteMem
  - CameraDefs Class, [103](#)
- GevSupportedOptionSelectorEnums
  - CameraDefs Class, [102](#)
- GevTimestampTickFrequency
  - Spinnaker::Camera, [486](#)
- GevTotalPacketCount
  - Spinnaker::TransportLayerStream, [842](#)
- GevVersionMajor
  - Spinnaker::TransportLayerDevice, [831](#)
- GevVersionMinor
  - Spinnaker::TransportLayerDevice, [831](#)
- GuiXmlManifestAddress
  - Spinnaker::Camera, [486](#)
- Guru
  - Types Enums, [314](#)
- H264Option, [664](#)
  - Spinnaker::Video::H264Option, [665](#)
- HEATMAP\_BLACK
  - Spinnaker Definitions, [163](#)
- HEATMAP\_BLUE
  - Spinnaker Definitions, [163](#)
- HEATMAP\_CYAN
  - Spinnaker Definitions, [163](#)
- HEATMAP\_GREEN
  - Spinnaker Definitions, [163](#)
- HEATMAP\_RED

- Spinnaker Definitions, [163](#)
- HEATMAP\_WHITE
  - Spinnaker Definitions, [163](#)
- HEATMAP\_YELLOW
  - Spinnaker Definitions, [163](#)
- HQ\_LINEAR
  - Spinnaker Definitions, [161](#)
- HUE
  - Spinnaker Definitions, [167](#)
- HasCRC
  - Spinnaker::GenApi::CChunkAdapterDcam, [533](#)
  - Spinnaker::Image, [690](#)
  - Spinnaker::Image, [719](#)
- HasInc
  - IFloat Interface, [250](#)
  - Spinnaker::GenApi::FloatNode, [649](#)
- Header
  - GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, [662](#)
  - GVCP\_EVENT\_REQUEST, [661](#)
  - GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID, [663](#)
  - GVCP\_EVENTDATA\_REQUEST, [662](#)
- HeatMapColor
  - Spinnaker Definitions, [163](#)
- Height
  - Spinnaker::Camera, [486](#)
- height
  - Spinnaker::Video::H264Option, [665](#)
- HeightMax
  - Spinnaker::Camera, [486](#)
- HexNumber
  - Types Enums, [313](#)
- HostAdapterDriverVersion
  - Spinnaker::TransportLayerInterface, [837](#)
- HostAdapterName
  - Spinnaker::TransportLayerInterface, [837](#)
- HostAdapterVendor
  - Spinnaker::TransportLayerInterface, [837](#)
- IArrivalEvent, [666](#)
  - Spinnaker::IArrivalEvent, [667](#)
- IBase
  - IBase Interface, [202](#)
- IBase Interface, [202](#)
  - IBase, [202](#)
- IBoolean
  - IBoolean Interface, [236](#)
- IBoolean Interface, [235](#)
  - GetValue, [235](#)
  - IBoolean, [236](#)
  - operator(), [235](#)
  - operator=, [236](#)
  - Verify, [236](#)
- ICameraBase, [667](#)
  - Spinnaker::ICameraBase, [669](#)
  - Spinnaker::TransportLayerDevice, [827](#)
  - Spinnaker::TransportLayerStream, [841](#)
- ICameraList, [672](#)
  - Spinnaker::ICameraList, [673](#)
- ICategory
  - ICategory Interfaces, [237](#)
- ICategory Interfaces, [237](#)
  - ICategory, [237](#)
- IChunkData, [675](#)
  - Spinnaker::IChunkData, [676](#)
- IChunkData Class, [184](#)
- IChunkPort
  - IChunkPort Interface, [239](#)
- IChunkPort Interface, [238](#)
  - CHUNK\_BASE\_ADDRESS\_REGISTER\_LEN, [238](#)
  - CHUNK\_BASE\_ADDRESS\_REGISTER, [238](#)
  - CHUNK\_LENGTH\_REGISTER\_LEN, [239](#)
  - CHUNK\_LENGTH\_REGISTER, [238](#)
  - CacheChunkData, [239](#)
  - IChunkPort, [239](#)
- ICommand
  - ICommand Interface, [240](#)
- ICommand Interface, [240](#)
  - ICommand, [240](#)
  - IsDone, [240](#)
- IDataStream
  - Spinnaker::Event, [637](#)
  - Spinnaker::Image, [723](#)
- IDestroy
  - IDestroy Interface, [241](#)
- IDestroy Interface, [241](#)
  - IDestroy, [241](#)
- IDevFileStream
  - Spinnaker::GenApi, [371](#)
- IDevFileStreamBase< CharType, Traits >, [680](#)
- IDevFileStreamBuf
  - Spinnaker::GenApi::IDevFileStreamBuf, [683](#)
- IDevFileStreamBuf< CharType, Traits >, [682](#)
- IDeviceEvent, [683](#)
  - Spinnaker::IDeviceEvent, [684](#)
- IDeviceInfo
  - IDeviceInfo Interface, [243](#)
- IDeviceInfo Interface, [242](#)
  - GetDeviceVersion, [242](#)
  - GetGenApiVersion, [242](#)
  - GetProductGuid, [243](#)
  - GetSchemaVersion, [243](#)
  - GetStandardNameSpace, [243](#)
  - GetToolTip, [243](#)
  - GetVendorName, [243](#)
  - GetVersionGuid, [243](#)
  - IDeviceInfo, [243](#)
- IEnumEntry
  - IEnumEntry Interface, [244](#)
- IEnumEntry Interface, [244](#)
  - GetNumericValue, [244](#)
  - GetSymbolic, [244](#)
  - IEnumEntry, [244](#)
  - IsSelfClearing, [244](#)
- IEnumReference
  - IEnumerationT Interface, [248](#)
- IEnumeration

- IEnumerator Interface, [246](#)
- IEnumerator Interface, [245](#)
  - GetCurrentEntry, [245](#)
  - GetEntries, [245](#)
  - GetEntry, [245](#)
  - GetEntryByName, [246](#)
  - GetIntValue, [246](#)
  - IEnumerator, [246](#)
  - operator\*, [246](#)
  - SetIntValue, [246](#)
- IEnumeratorT Interface, [247](#)
  - GetEntry, [247](#)
  - IEnumReference, [248](#)
  - IEnumeratorT, [248](#)
  - operator=, [247](#)
- IEnumeratorT
  - IEnumeratorT Interface, [248](#)
- IFloat
  - IFloat Interface, [251](#)
- IFloat Interface, [249](#)
  - GetDisplayNotation, [250](#)
  - GetDisplayPrecision, [250](#)
  - GetInc, [250](#)
  - GetIncMode, [250](#)
  - GetListOfValidValues, [250](#)
  - GetMax, [250](#)
  - GetMin, [250](#)
  - GetRepresentation, [250](#)
  - GetUnit, [250](#)
  - HasInc, [250](#)
  - IFloat, [251](#)
  - ImposeMax, [251](#)
  - ImposeMin, [251](#)
  - operator=, [251](#)
- IIDC
  - Types Enums, [314](#)
- Image, [685](#)
  - Spinnaker::Image, [687](#)
- Image Class, [185](#)
- ImageEvent, [692](#)
  - Spinnaker::ImageEvent, [693](#)
- ImageStatistics, [693](#)
  - Spinnaker::ImageStatistics, [694](#)
- ImageStatistics Class, [186](#)
- Integer
  - Integer Interface, [252](#)
- Integer Interface, [252](#)
  - Integer, [252](#)
  - ImposeMax, [252](#)
  - ImposeMin, [252](#)
  - operator=, [252](#)
- Interface, [696](#)
  - Spinnaker::Interface, [697](#)
  - Spinnaker::TransportLayerInterface, [834](#)
- Interface Class, [187](#)
- InterfaceEvent, [699](#)
  - Spinnaker::InterfaceEvent, [700](#)
- InterfaceList, [700](#)
  - Spinnaker::InterfaceList, [701](#)
- InterfaceList Class, [188](#)
- ILoggingEvent, [702](#)
  - Spinnaker::ILoggingEvent, [703](#)
- IMAGE\_CHUNK\_DATA\_INVALID
  - Spinnaker Definitions, [164](#)
- IMAGE\_CRC\_CHECK\_FAILED
  - Spinnaker Definitions, [164](#)
- IMAGE\_DATA\_INCOMPLETE
  - Spinnaker Definitions, [164](#)
- IMAGE\_DATA\_OVERFLOW
  - Spinnaker Definitions, [164](#)
- IMAGE\_FILE\_FORMAT\_FORCE\_32BITS
  - Spinnaker Definitions, [163](#)
- IMAGE\_INFO\_INCONSISTENT
  - Spinnaker Definitions, [164](#)
- IMAGE\_LEADER\_BUFFER\_SIZE\_INCONSISTENT
  - Spinnaker Definitions, [164](#)
- IMAGE\_MISSING\_LEADER
  - Spinnaker Definitions, [164](#)
- IMAGE\_MISSING\_PACKETS
  - Spinnaker Definitions, [164](#)
- IMAGE\_MISSING\_TRAILER
  - Spinnaker Definitions, [164](#)
- IMAGE\_NO\_ERROR
  - Spinnaker Definitions, [164](#)
- IMAGE\_NO\_SYSTEM\_RESOURCES
  - Spinnaker Definitions, [164](#)
- IMAGE\_PACKETID\_INCONSISTENT
  - Spinnaker Definitions, [164](#)
- IMAGE\_TRAILER\_BUFFER\_SIZE\_INCONSISTENT
  - Spinnaker Definitions, [164](#)
- IMAGE\_UNKNOWN\_ERROR
  - Spinnaker Definitions, [164](#)
- INTEGRAL\_CAST2
  - GCUtilities Utility, [233](#)
- INTEGRAL\_CAST
  - GCUtilities Utility, [233](#)
- INode
  - INode Interface, [259](#)
- INode Interface, [253](#)
  - Combine, [255](#)
  - DeregisterCallback, [255](#)
  - GetAlias, [255](#)
  - GetCachingMode, [255](#)
  - GetCastAlias, [255](#)
  - GetChildren, [255](#)
  - GetDescription, [256](#)
  - GetDisplayName, [256](#)
  - GetDocuURL, [256](#)
  - GetEventID, [256](#)
  - GetNameSpace, [256](#)
  - GetNodeMap, [256](#)
  - GetParents, [256](#)
  - GetPollingTime, [256](#)
  - GetPrincipalInterfaceType, [256](#)
  - GetProperty, [257](#)
  - GetPropertyNames, [257](#)

- GetVisibility, [257](#)
- INode, [259](#)
- IRReference, [259](#)
- ImposeAccessMode, [257](#)
- ImposeVisibility, [257](#)
- InvalidateNode, [257](#)
- IsAccessModeCacheable, [257](#)
- IsAvailable, [257](#), [258](#)
- IsCacheable, [258](#)
- IsCacheable, [258](#)
- IsDeprecated, [258](#)
- IsFeature, [258](#)
- IsImplemented, [258](#)
- IsReadable, [258](#), [259](#)
- IsStreamable, [259](#)
- IsVisible, [259](#)
- IsWritable, [259](#)
- operator!=, [259](#)
- operator==, [259](#)
- RegisterCallback, [259](#)
- INodeMap
  - INodeMap Interface, [261](#)
- INodeMap Interface, [260](#)
  - Connect, [260](#)
  - GetDeviceName, [261](#)
  - GetLock, [261](#)
  - GetNode, [261](#)
  - GetNumNodes, [261](#)
  - INodeMap, [261](#)
  - InvalidateNodes, [261](#)
  - Poll, [261](#)
- INodeMapDyn
  - INodeMapDyn Interface, [265](#)
- INodeMapDyn Interface, [262](#)
  - ExtractIndependentSubtree, [263](#)
  - GetSupportedSchemaVersions, [263](#)
  - INodeMapDyn, [265](#)
  - LoadXMLFromFile, [263](#)
  - LoadXMLFromFileInject, [263](#)
  - LoadXMLFromString, [263](#)
  - LoadXMLFromStringInject, [263](#)
  - LoadXMLFromZIPData, [264](#)
  - LoadXMLFromZIPFile, [264](#)
  - MergeXMLFiles, [264](#)
  - PreprocessXMLFromFile, [264](#)
  - PreprocessXMLFromZIPFile, [265](#)
- IPV4Address
  - Types Enums, [313](#)
- IPersistScript
  - Spinnaker::GenApi, [372](#)
- IPort
  - IPort Interface, [268](#)
- IPort Interface, [268](#)
  - Address, [268](#)
  - IPort, [268](#)
  - Length, [268](#)
  - Write, [268](#)
- IPortConstruct
  - IPortConstruct Interface, [269](#)
- IPortConstruct Interface, [269](#)
  - GetSwapEndianness, [269](#)
  - IPortConstruct, [269](#)
- IPortRecorder
  - IPortRecorder Interface, [271](#)
- IPortRecorder Interface, [270](#)
  - GetCookie, [270](#)
  - IPortRecorder, [271](#)
  - IPortReplay, [271](#)
  - IPortWriteList, [271](#)
  - Invalidate, [271](#)
  - Replay, [270](#)
  - SetCookie, [270](#)
  - StopRecording, [271](#)
- IPortReplay
  - IPortRecorder Interface, [271](#)
- IPortWriteList
  - IPortRecorder Interface, [271](#)
- IPP
  - Spinnaker Definitions, [161](#)
- IRReference
  - INode Interface, [259](#)
- IRRegister
  - IRRegister Interfaces, [273](#)
- IRRegister Interfaces, [272](#)
  - Get, [272](#)
  - GetAddress, [272](#)
  - GetLength, [273](#)
  - IRRegister, [273](#)
- IRemovalEvent, [752](#)
  - Spinnaker::IRemovalEvent, [753](#)
- ISelector
  - ISelector Interface, [274](#)
- ISelector Interface, [274](#)
  - GetSelectedFeatures, [274](#)
  - GetSelectingFeatures, [274](#)
  - ISelector, [274](#)
- ISelectorDigit
  - ISelectorDigit Interface, [276](#)
- ISelectorDigit Interface, [275](#)
  - GetSelectorList, [275](#)
  - ISelectorDigit, [276](#)
  - Restore, [275](#)
  - SetNext, [276](#)
  - ToString, [276](#)
- IStrng
  - IStrng Class, [277](#)
- IStrng Class, [277](#)
  - GetMaxLength, [277](#)
  - IStrng, [277](#)
- ISystem, [753](#)
  - Spinnaker::ISystem, [754](#)
- ISystem Class, [189](#)
- IValue
  - IValue Class, [279](#)
- IValue Class, [278](#)
  - FromString, [278](#)

- IValue, [279](#)
  - IsValueCacheValid, [278](#)
  - ToString, [278](#)
- idFrom
  - Types Enums, [312](#)
- idNone
  - Types Enums, [312](#)
- idTo
  - Types Enums, [312](#)
- Image, [704](#)
  - Spinnaker::Image, [707](#), [708](#)
- Image Class, [142](#)
- ImageComponentEnable
  - Spinnaker::Camera, [486](#)
- ImageComponentSelector
  - Spinnaker::Camera, [486](#)
- ImageComponentSelector\_Color
  - CameraDefs Class, [103](#)
- ImageComponentSelector\_Confidence
  - CameraDefs Class, [103](#)
- ImageComponentSelector\_Disparity
  - CameraDefs Class, [103](#)
- ImageComponentSelector\_Infrared
  - CameraDefs Class, [103](#)
- ImageComponentSelector\_Intensity
  - CameraDefs Class, [103](#)
- ImageComponentSelector\_Range
  - CameraDefs Class, [103](#)
- ImageComponentSelector\_Scatter
  - CameraDefs Class, [104](#)
- ImageComponentSelector\_Ultraviolet
  - CameraDefs Class, [103](#)
- ImageComponentSelectorEnums
  - CameraDefs Class, [103](#)
- ImageCompressionBitrate
  - Spinnaker::Camera, [487](#)
- ImageCompressionJPEGFormatOption
  - Spinnaker::Camera, [487](#)
- ImageCompressionJPEGFormatOption\_Baseline↵
  - Optimized
  - CameraDefs Class, [104](#)
- ImageCompressionJPEGFormatOption\_Baseline↵
  - Standard
  - CameraDefs Class, [104](#)
- ImageCompressionJPEGFormatOption\_Lossless
  - CameraDefs Class, [104](#)
- ImageCompressionJPEGFormatOption\_Progressive
  - CameraDefs Class, [104](#)
- ImageCompressionJPEGFormatOptionEnums
  - CameraDefs Class, [104](#)
- ImageCompressionMode
  - Spinnaker::Camera, [487](#)
- ImageCompressionMode\_Lossless
  - CameraDefs Class, [104](#)
- ImageCompressionMode\_Off
  - CameraDefs Class, [104](#)
- ImageCompressionModeEnums
  - CameraDefs Class, [104](#)
- ImageCompressionQuality
  - Spinnaker::Camera, [487](#)
- ImageCompressionRateOption
  - Spinnaker::Camera, [487](#)
- ImageCompressionRateOption\_FixBitrate
  - CameraDefs Class, [104](#)
- ImageCompressionRateOption\_FixQuality
  - CameraDefs Class, [104](#)
- ImageCompressionRateOptionEnums
  - CameraDefs Class, [104](#)
- ImageConverter
  - Spinnaker::Image, [723](#)
- ImageEvent, [724](#)
  - Spinnaker::ImageEvent, [725](#)
- ImageEvent Class, [143](#)
- ImageFileFormat
  - Spinnaker Definitions, [163](#)
- ImageFiler
  - Spinnaker::Image, [723](#)
- ImagePtr, [726](#)
  - Spinnaker::ImagePtr, [727](#)
- ImagePtr Class, [144](#)
- ImageStatistics, [728](#)
  - Spinnaker::ImageStatistics, [729](#)
- ImageStatistics Class, [145](#)
- ImageStatsCalculator
  - Spinnaker::Image, [723](#)
  - Spinnaker::ImageStatistics, [733](#)
- ImageStatus
  - Spinnaker Definitions, [163](#)
- ImposeAccessMode
  - INode Interface, [257](#)
  - Spinnaker::GenApi::Node, [776](#)
- ImposeMax
  - IFloat Interface, [251](#)
  - Integer Interface, [252](#)
  - Spinnaker::GenApi::FloatNode, [649](#)
  - Spinnaker::GenApi::IntegerNode, [738](#)
- ImposeMin
  - IFloat Interface, [251](#)
  - Integer Interface, [252](#)
  - Spinnaker::GenApi::FloatNode, [649](#)
  - Spinnaker::GenApi::IntegerNode, [738](#)
- ImposeVisibility
  - INode Interface, [257](#)
  - Spinnaker::GenApi::Node, [776](#)
- include/AVIRecorder.h, [853](#)
- include/ArrivalEvent.h, [851](#)
- include/BasePtr.h, [853](#)
- include/Camera.h, [855](#)
- include/CameraBase.h, [857](#)
- include/CameraDefs.h, [859](#)
- include/CameraList.h, [889](#)
- include/CameraPtr.h, [891](#)
- include/ChunkData.h, [893](#)
- include/DeviceEvent.h, [895](#)
- include/Event.h, [897](#)
- include/Exception.h, [899](#)

include/Image.h, 900  
include/ImageEvent.h, 902  
include/ImagePtr.h, 903  
include/ImageStatistics.h, 905  
include/Interface.h, 906  
include/Interface/IArrivalEvent.h, 908  
include/Interface/ICameraBase.h, 910  
include/Interface/ICameraList.h, 912  
include/Interface/IChunkData.h, 914  
include/Interface/IDeviceEvent.h, 916  
include/Interface/ILmage.h, 918  
include/Interface/ILmageEvent.h, 920  
include/Interface/ILmageStatistics.h, 922  
include/Interface/IInterface.h, 924  
include/Interface/IInterfaceEvent.h, 926  
include/Interface/IInterfaceList.h, 928  
include/Interface/ILoggingEvent.h, 929  
include/Interface/IRemovalEvent.h, 931  
include/Interface/ISystem.h, 933  
include/InterfaceEvent.h, 934  
include/InterfaceList.h, 936  
include/InterfacePtr.h, 937  
include/LoggingEvent.h, 939  
include/LoggingEventData.h, 940  
include/LoggingEventDataPtr.h, 942  
include/RemovalEvent.h, 944  
include/SpinGenApi/Autovector.h, 946  
include/SpinGenApi/Base.h, 947  
include/SpinGenApi/BooleanNode.h, 948  
include/SpinGenApi/CategoryNode.h, 950  
include/SpinGenApi/ChunkAdapter.h, 952  
include/SpinGenApi/ChunkAdapterDcam.h, 954  
include/SpinGenApi/ChunkAdapterGEV.h, 958  
include/SpinGenApi/ChunkAdapterGeneric.h, 956  
include/SpinGenApi/ChunkAdapterU3V.h, 960  
include/SpinGenApi/ChunkPort.h, 962  
include/SpinGenApi/CommandNode.h, 964  
include/SpinGenApi/Compatibility.h, 967  
include/SpinGenApi/Container.h, 968  
include/SpinGenApi/Counter.h, 968  
include/SpinGenApi/EnumClasses.h, 969  
include/SpinGenApi/EnumEntryNode.h, 971  
include/SpinGenApi/EnumNode.h, 973  
include/SpinGenApi/EnumNodeT.h, 975  
include/SpinGenApi/EventAdapter.h, 977  
include/SpinGenApi/EventAdapter1394.h, 979  
include/SpinGenApi/EventAdapterGEV.h, 983  
include/SpinGenApi/EventAdapterGeneric.h, 981  
include/SpinGenApi/EventAdapterU3V.h, 985  
include/SpinGenApi/EventPort.h, 987  
include/SpinGenApi/Filestream.h, 989  
include/SpinGenApi/FloatNode.h, 991  
include/SpinGenApi/FloatRegNode.h, 993  
include/SpinGenApi/GCBase.h, 995  
include/SpinGenApi/GCString.h, 996  
include/SpinGenApi/GCStringVector.h, 998  
include/SpinGenApi/GCSynch.h, 998  
include/SpinGenApi/GCTypes.h, 999  
include/SpinGenApi/GCUtilities.h, 1001  
include/SpinGenApi/IBoolean.h, 1005  
include/SpinGenApi/ICategory.h, 1007  
include/SpinGenApi/IChunkPort.h, 1009  
include/SpinGenApi/ICommand.h, 1011  
include/SpinGenApi/IDestroy.h, 1013  
include/SpinGenApi/DeviceInfo.h, 1015  
include/SpinGenApi/IEnumEntry.h, 1017  
include/SpinGenApi/IEnumeration.h, 1019  
include/SpinGenApi/IEnumerationT.h, 1021  
include/SpinGenApi/IFloat.h, 1023  
include/SpinGenApi/IInteger.h, 1025  
include/SpinGenApi/INode.h, 1027  
include/SpinGenApi/INodeMap.h, 1030  
include/SpinGenApi/INodeMapDyn.h, 1032  
include/SpinGenApi/IPort.h, 1038  
include/SpinGenApi/IPortConstruct.h, 1039  
include/SpinGenApi/IPortRecorder.h, 1041  
include/SpinGenApi/IRegister.h, 1043  
include/SpinGenApi/ISelector.h, 1045  
include/SpinGenApi/ISelectorDigit.h, 1046  
include/SpinGenApi/IString.h, 1048  
include/SpinGenApi/IValue.h, 1050  
include/SpinGenApi/IntRegNode.h, 1036  
include/SpinGenApi/IntegerNode.h, 1034  
include/SpinGenApi/Node.h, 1051  
include/SpinGenApi/NodeCallback.h, 1053  
include/SpinGenApi/NodeCallbackImpl.h, 1055  
include/SpinGenApi/NodeMap.h, 1056  
include/SpinGenApi/NodeMapFactory.h, 1058  
include/SpinGenApi/NodeMapRef.h, 1059  
include/SpinGenApi/Persistence.h, 1060  
include/SpinGenApi/Pointer.h, 1062  
include/SpinGenApi/PortImpl.h, 1065  
include/SpinGenApi/PortNode.h, 1066  
include/SpinGenApi/PortRecorder.h, 1068  
include/SpinGenApi/PortReplay.h, 1069  
include/SpinGenApi/PortWriteList.h, 1070  
include/SpinGenApi/Reference.h, 1072  
include/SpinGenApi/RegisterNode.h, 1073  
include/SpinGenApi/RegisterPortImpl.h, 1075  
include/SpinGenApi/SelectorSet.h, 1075  
include/SpinGenApi/SpinTestCamera.h, 1078  
include/SpinGenApi/SpinnakerGenApi.h, 1076  
include/SpinGenApi/StringNode.h, 1078  
include/SpinGenApi/StringRegNode.h, 1080  
include/SpinGenApi/StructPort.h, 1082  
include/SpinGenApi/Synch.h, 1082  
include/SpinGenApi/Types.h, 1083  
include/SpinGenApi/ValueNode.h, 1087  
include/SpinUpdate.h, 1095  
include/SpinVideo.h, 1097  
include/SpinVideoDefs.h, 1097  
include/Spinnaker.h, 1089  
include/SpinnakerDefs.h, 1090  
include/SpinnakerPlatform.h, 1095  
include/System.h, 1098  
include/SystemPtr.h, 1100



- include/TransportLayerDefs.h, [1102](#)
- include/TransportLayerDevice.h, [1104](#)
- include/TransportLayerInterface.h, [1106](#)
- include/TransportLayerStream.h, [1108](#)
- IncompatibleDeviceCount
  - Spinnaker::TransportLayerInterface, [837](#)
- IncompatibleDeviceID
  - Spinnaker::TransportLayerInterface, [837](#)
- IncompatibleDeviceModelName
  - Spinnaker::TransportLayerInterface, [837](#)
- IncompatibleDeviceSelector
  - Spinnaker::TransportLayerInterface, [838](#)
- IncompatibleDeviceVendorName
  - Spinnaker::TransportLayerInterface, [838](#)
- IncompatibleGevDeviceIPAddress
  - Spinnaker::TransportLayerInterface, [838](#)
- IncompatibleGevDeviceMACAddress
  - Spinnaker::TransportLayerInterface, [838](#)
- IncompatibleGevDeviceSubnetMask
  - Spinnaker::TransportLayerInterface, [838](#)
- Increasing
  - Types Enums, [314](#)
- indexedColor\_8bit
  - Spinnaker::BMPOption, [383](#)
- Init
  - Spinnaker::Camera, [416](#)
  - Spinnaker::CameraBase, [519](#)
  - Spinnaker::ICameraBase, [670](#)
- int64\_autovector\_t, [733](#)
  - Spinnaker::GenApi::int64\_autovector\_t, [734](#)
- IntRegNode, [750](#)
  - Spinnaker::GenApi::IntRegNode, [751](#)
- IntRegNode Class, [267](#)
- IntType\_FLOAT32
  - Spinnaker Definitions, [165](#)
- IntType\_INT8
  - Spinnaker Definitions, [165](#)
- IntType\_UINT10
  - Spinnaker Definitions, [165](#)
- IntType\_UINT10P
  - Spinnaker Definitions, [165](#)
- IntType\_UINT10p
  - Spinnaker Definitions, [165](#)
- IntType\_UINT12
  - Spinnaker Definitions, [165](#)
- IntType\_UINT12P
  - Spinnaker Definitions, [165](#)
- IntType\_UINT12p
  - Spinnaker Definitions, [165](#)
- IntType\_UINT14
  - Spinnaker Definitions, [165](#)
- IntType\_UINT16
  - Spinnaker Definitions, [165](#)
- IntType\_UINT8
  - Spinnaker Definitions, [165](#)
- IntType\_UNKNOWN
  - Spinnaker Definitions, [165](#)
- IntegerNode, [735](#)
  - Spinnaker::GenApi::IntegerNode, [737](#)
- IntegerNode Class, [266](#)
  - CIntegerRef, [266](#)
- Interface, [739](#)
  - Spinnaker::TransportLayerInterface, [834](#)
- interface
  - Types.h, [1086](#)
- Interface Class, [146](#)
- InterfaceDisplayName
  - Spinnaker::TransportLayerInterface, [838](#)
- InterfaceEvent, [743](#)
  - Spinnaker::InterfaceEvent, [745](#)
- InterfaceEvent Class, [147](#)
- InterfaceID
  - Spinnaker::TransportLayerInterface, [838](#)
- InterfaceImpl
  - Spinnaker::CameraBase, [521](#)
  - Spinnaker::ICameraBase, [672](#)
  - Spinnaker::ICameraList, [674](#)
- InterfaceInternal
  - Spinnaker::IInterface, [698](#)
  - Spinnaker::Interface, [743](#)
  - Spinnaker::TransportLayerInterface, [834](#)
- InterfaceList, [745](#)
  - Spinnaker::InterfaceList, [747](#)
- InterfaceList Class, [148](#)
- InterfacePtr, [748](#)
  - Spinnaker::InterfacePtr, [749](#)
- InterfacePtr Class, [149](#)
- InterfaceType
  - Spinnaker::TransportLayerInterface, [839](#)
- interlaced
  - Spinnaker::PNGOption, [790](#)
- intfIBase
  - Types Enums, [312](#)
- intfIBoolean
  - Types Enums, [312](#)
- intfICategory
  - Types Enums, [312](#)
- intfICommand
  - Types Enums, [312](#)
- intfIEnumEntry
  - Types Enums, [312](#)
- intfIEnumeration
  - Types Enums, [312](#)
- intfIFloat
  - Types Enums, [312](#)
- intfIInteger
  - Types Enums, [312](#)
- intfIIPort
  - Types Enums, [312](#)
- intfIRegister
  - Types Enums, [312](#)
- intfIString
  - Types Enums, [312](#)
- intfIValue
  - Types Enums, [312](#)
- InvalidDate

- IPortRecorder Interface, [271](#)
- InvalidateNode
  - INode Interface, [257](#)
  - Spinnaker::GenApi::CChunkPort, [541](#)
  - Spinnaker::GenApi::CEventPort, [557](#)
  - Spinnaker::GenApi::CPortImpl, [604](#)
  - Spinnaker::GenApi::Node, [776](#)
- InvalidateNodes
  - INodeMap Interface, [261](#)
  - Spinnaker::GenApi::NodeMap, [783](#)
- InverseChunkLength
  - DCAM\_CHUNK\_TRAILER, [615](#)
- Invisible
  - Types Enums, [314](#)
- ios\_type
  - Spinnaker::GenApi::IDevFileStreamBase, [681](#)
  - Spinnaker::GenApi::ODevFileStreamBase, [786](#)
- is\_open
  - Spinnaker::GenApi::IDevFileStreamBase, [681](#)
  - Spinnaker::GenApi::IDevFileStreamBuf, [683](#)
  - Spinnaker::GenApi::ODevFileStreamBase, [786](#)
  - Spinnaker::GenApi::ODevFileStreamBuf, [788](#)
- IsAccessModeCacheable
  - INode Interface, [257](#)
  - Spinnaker::GenApi::Node, [776](#)
- IsAvailable
  - INode Interface, [257](#), [258](#)
  - Pointer Class, [291](#)
- IsCachable
  - INode Interface, [258](#)
  - Spinnaker::GenApi::Node, [776](#)
- IsCacheable
  - INode Interface, [258](#)
- IsCameraDescriptionFileDataReleased
  - Spinnaker::GenApi::CNodeMapFactory, [587](#)
- IsDeprecated
  - INode Interface, [258](#)
  - Spinnaker::GenApi::Node, [776](#)
- IsDone
  - ICommand Interface, [240](#)
  - Spinnaker::GenApi::CommandNode, [597](#)
- IsEmpty
  - Spinnaker::GenApi::CNodeMapFactory, [588](#)
  - Spinnaker::GenApi::CSelectorSet, [611](#)
- IsFeature
  - INode Interface, [258](#)
  - Spinnaker::GenApi::Node, [776](#)
- IsImplemented
  - INode Interface, [258](#)
  - Pointer Class, [292](#)
- IsInUse
  - Spinnaker::Image, [690](#)
  - Spinnaker::IInterface, [698](#)
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::Image, [719](#)
  - Spinnaker::Interface, [741](#)
  - Spinnaker::System, [818](#)
- IsIncomplete
  - Spinnaker::Image, [690](#)
  - Spinnaker::Image, [719](#)
- IsInitialized
  - Spinnaker::CameraBase, [519](#)
  - Spinnaker::ICameraBase, [670](#)
- IsLoaded
  - Spinnaker::GenApi::CNodeMapFactory, [588](#)
- IsPreprocessed
  - Spinnaker::GenApi::CNodeMapFactory, [588](#)
- IsReadable
  - INode Interface, [258](#), [259](#)
  - Pointer Class, [292](#)
- IsSelector
  - Spinnaker::GenApi::Node, [777](#)
- IsSelfClearing
  - IEnumEntry Interface, [244](#)
  - Spinnaker::GenApi::EnumEntryNode, [627](#)
- IsStreamable
  - INode Interface, [259](#)
  - Spinnaker::GenApi::Node, [777](#)
- IsStreaming
  - Spinnaker::CameraBase, [519](#)
  - Spinnaker::ICameraBase, [671](#)
- IsValid
  - Spinnaker::BasePtr, [381](#)
  - Spinnaker::CameraBase, [520](#)
  - Spinnaker::GenApi::CPointer, [600](#)
  - Spinnaker::GenICam::CGlobalLock, [563](#)
  - Spinnaker::ICameraBase, [671](#)
- IsValueCacheValid
  - IValue Class, [278](#)
  - Spinnaker::GenApi::ValueNode, [849](#)
- IsVisible
  - INode Interface, [259](#)
- IsWritable
  - INode Interface, [259](#)
  - Pointer Class, [292](#)
- IsZero
  - Spinnaker::GenApi::Counter, [598](#)
- IspEnable
  - Spinnaker::Camera, [487](#)
- istream\_type
  - Spinnaker::GenApi::IDevFileStreamBase, [681](#)
- Items
  - GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, [662](#)
  - GVCP\_EVENT\_REQUEST, [661](#)
- JPEG12\_C
  - Spinnaker Definitions, [163](#)
- JPEG2000
  - Spinnaker Definitions, [163](#)
- JPEGOOption, [756](#)
  - Spinnaker::JPEGOOption, [756](#)
- JPEG
  - Spinnaker Definitions, [163](#)
  - Spinnaker::TIFFOption, [824](#)
- JPG2Option, [757](#)
  - Spinnaker::JPG2Option, [757](#)



- LIGHTNESS
  - Spinnaker Definitions, [167](#)
- LOG\_LEVEL\_ALERT
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_CRIT
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_DEBUG
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_ERROR
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_FATAL
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_INFO
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_NOTICE
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_NOTSET
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_OFF
  - Spinnaker Definitions, [166](#)
- LOG\_LEVEL\_WARN
  - Spinnaker Definitions, [166](#)
- LUTEnable
  - Spinnaker::Camera, [490](#)
- LUTIndex
  - Spinnaker::Camera, [490](#)
- LUTSelector
  - Spinnaker::Camera, [490](#)
- LUTSelector\_LUT1
  - CameraDefs Class, [108](#)
- LUTSelectorEnums
  - CameraDefs Class, [107](#)
- LUTValue
  - Spinnaker::Camera, [490](#)
- LUTValueAll
  - Spinnaker::Camera, [491](#)
- LZW
  - Spinnaker::TIFFOption, [824](#)
- Length
  - GVCP\_REQUEST\_HEADER, [664](#)
  - IPort Interface, [268](#)
  - U3V\_COMMAND\_HEADER, [845](#)
- length
  - Spinnaker::GenICam::gcstring, [656](#)
- LibraryVersion, [758](#)
- LineFilterWidth
  - Spinnaker::Camera, [487](#)
- LineFormat
  - Spinnaker::Camera, [488](#)
- LineFormat\_LVDS
  - CameraDefs Class, [105](#)
- LineFormat\_NoConnect
  - CameraDefs Class, [105](#)
- LineFormat\_OpenDrain
  - CameraDefs Class, [105](#)
- LineFormat\_OptoCoupled
  - CameraDefs Class, [105](#)
- LineFormat\_RS422
  - CameraDefs Class, [105](#)
- LineFormat\_TTL
  - CameraDefs Class, [105](#)
- LineFormat\_TriState
  - CameraDefs Class, [105](#)
- LineFormatEnums
  - CameraDefs Class, [104](#)
- LineInputFilterSelector
  - Spinnaker::Camera, [488](#)
- LineInputFilterSelector\_Debounce
  - CameraDefs Class, [105](#)
- LineInputFilterSelector\_Deg glitch
  - CameraDefs Class, [105](#)
- LineInputFilterSelectorEnums
  - CameraDefs Class, [105](#)
- LineInverter
  - Spinnaker::Camera, [488](#)
- LineMode
  - Spinnaker::Camera, [488](#)
- LineMode\_Input
  - CameraDefs Class, [105](#)
- LineMode\_Output
  - CameraDefs Class, [105](#)
- LineModeEnums
  - CameraDefs Class, [105](#)
- LinePitch
  - Spinnaker::Camera, [488](#)
- LineSelector
  - Spinnaker::Camera, [488](#)
- LineSelector\_Line0
  - CameraDefs Class, [105](#)
- LineSelector\_Line1
  - CameraDefs Class, [105](#)
- LineSelector\_Line2
  - CameraDefs Class, [105](#)
- LineSelector\_Line3
  - CameraDefs Class, [105](#)
- LineSelectorEnums
  - CameraDefs Class, [105](#)
- LineSource
  - Spinnaker::Camera, [488](#)
- LineSource\_AllPixel
  - CameraDefs Class, [106](#)
- LineSource\_AnyPixel
  - CameraDefs Class, [106](#)
- LineSource\_Counter0Active
  - CameraDefs Class, [106](#)
- LineSource\_Counter1Active
  - CameraDefs Class, [106](#)
- LineSource\_ExposureActive
  - CameraDefs Class, [106](#)
- LineSource\_FrameTriggerWait
  - CameraDefs Class, [106](#)
- LineSource\_Line0
  - CameraDefs Class, [106](#)
- LineSource\_Line1
  - CameraDefs Class, [106](#)
- LineSource\_Line2

- CameraDefs Class, [106](#)
- LineSource\_Line3
  - CameraDefs Class, [106](#)
- LineSource\_LogicBlock0
  - CameraDefs Class, [106](#)
- LineSource\_LogicBlock1
  - CameraDefs Class, [106](#)
- LineSource\_Off
  - CameraDefs Class, [106](#)
- LineSource\_PPSSignal
  - CameraDefs Class, [106](#)
- LineSource\_SerialPort0
  - CameraDefs Class, [106](#)
- LineSource\_UserOutput0
  - CameraDefs Class, [106](#)
- LineSource\_UserOutput1
  - CameraDefs Class, [106](#)
- LineSource\_UserOutput2
  - CameraDefs Class, [106](#)
- LineSource\_UserOutput3
  - CameraDefs Class, [106](#)
- LineSourceEnums
  - CameraDefs Class, [105](#)
- LineStatus
  - Spinnaker::Camera, [488](#)
- LineStatusAll
  - Spinnaker::Camera, [489](#)
- Linear
  - Types Enums, [313](#)
- LinkErrorCount
  - Spinnaker::Camera, [489](#)
- LinkUptime
  - Spinnaker::Camera, [489](#)
- listIncrement
  - Types Enums, [312](#)
- LittleEndian
  - Types Enums, [311](#)
- LoadAndInject
  - Spinnaker::GenApi::CNodeMapFactory, [588](#)
- LoadFromBag
  - Spinnaker::GenApi::CFeatureBag, [559](#)
- LoadXMLFromFile
  - INodeMapDyn Interface, [263](#)
  - Spinnaker::GenApi::NodeMap, [783](#)
- LoadXMLFromFileInject
  - INodeMapDyn Interface, [263](#)
  - Spinnaker::GenApi::NodeMap, [783](#)
- LoadXMLFromString
  - INodeMapDyn Interface, [263](#)
  - Spinnaker::GenApi::NodeMap, [783](#)
- LoadXMLFromStringInject
  - INodeMapDyn Interface, [263](#)
  - Spinnaker::GenApi::NodeMap, [783](#)
- LoadXMLFromZIPData
  - INodeMapDyn Interface, [264](#)
  - Spinnaker::GenApi::NodeMap, [783](#)
- LoadXMLFromZIPFile
  - INodeMapDyn Interface, [264](#)
- Spinnaker::GenApi::NodeMap, [783](#)
- Lock
  - Spinnaker::GenApi::CLock, [576](#)
  - Spinnaker::GenICam::CGlobalLock, [563](#)
  - Spinnaker::GenICam::CLock, [578](#)
  - Spinnaker::GenICam::LockableObject, [761](#)
  - Spinnaker::GenICam::LockableObject::Lock, [759](#)
- LockableObject< Object >, [760](#)
- LockableObject< Object >::Lock, [759](#)
- Logarithmic
  - Types Enums, [313](#)
- Logging Event Class, [151](#)
- LoggingEvent, [761](#)
  - Spinnaker::LoggingEvent, [762](#)
- LoggingEvent Class, [150](#)
- LoggingEventData, [763](#)
  - Spinnaker::LoggingEventData, [764](#)
- LoggingEventDataPtr, [766](#)
  - Spinnaker::LoggingEventDataPtr, [767](#)
- LoggingEventDataPtr Class, [152](#)
- LogicBlockLUTInputActivation
  - Spinnaker::Camera, [489](#)
- LogicBlockLUTInputActivation\_AnyEdge
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputActivation\_FallingEdge
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputActivation\_LevelHigh
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputActivation\_LevelLow
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputActivation\_RisingEdge
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputActivationEnums
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputSelector
  - Spinnaker::Camera, [489](#)
- LogicBlockLUTInputSelector\_Input0
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputSelector\_Input1
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputSelector\_Input2
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputSelector\_Input3
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputSelectorEnums
  - CameraDefs Class, [106](#)
- LogicBlockLUTInputSource
  - Spinnaker::Camera, [489](#)
- LogicBlockLUTInputSource\_AcquisitionActive
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Counter0End
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Counter0Start
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Counter1End
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Counter1Start
  - CameraDefs Class, [107](#)

- LogicBlockLUTInputSource\_ExposureEnd
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_ExposureStart
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_FrameTriggerWait
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Line0
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Line1
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Line2
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Line3
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_LogiBlock0
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_LogiBlock1
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_UserOutput0
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_UserOutput1
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_UserOutput2
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_UserOutput3
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSource\_Zero
  - CameraDefs Class, [107](#)
- LogicBlockLUTInputSourceEnums
  - CameraDefs Class, [106](#)
- LogicBlockLUTOutputValue
  - Spinnaker::Camera, [489](#)
- LogicBlockLUTOutputValueAll
  - Spinnaker::Camera, [489](#)
- LogicBlockLUTRowIndex
  - Spinnaker::Camera, [490](#)
- LogicBlockLUTSelector
  - Spinnaker::Camera, [490](#)
- LogicBlockLUTSelector\_Enable
  - CameraDefs Class, [107](#)
- LogicBlockLUTSelector\_Value
  - CameraDefs Class, [107](#)
- LogicBlockLUTSelectorEnums
  - CameraDefs Class, [107](#)
- LogicBlockSelector
  - Spinnaker::Camera, [490](#)
- LogicBlockSelector\_LogiBlock0
  - CameraDefs Class, [107](#)
- LogicBlockSelector\_LogiBlock1
  - CameraDefs Class, [107](#)
- LogicBlockSelectorEnums
  - CameraDefs Class, [107](#)
- m\_BaseAddress
  - Spinnaker::GenApi::CTestPortStruct, [614](#)
- m\_CallbackType
  - Spinnaker::GenApi::CNodeCallback, [581](#)
- m\_Callbacks
  - Spinnaker::GenApi::Node, [777](#)
- m\_DebugCount
  - Spinnaker::GenICam::CGlobalLock, [564](#)
- m\_Lock
  - Spinnaker::GenICam::CGlobalLockUnlocker, [565](#)
  - Spinnaker::GenICam::LockableObject, [761](#)
- m\_NumReads
  - Spinnaker::GenApi::CTestPortStruct, [614](#)
- m\_NumWrites
  - Spinnaker::GenApi::CTestPortStruct, [614](#)
- m\_bOwnLock
  - Spinnaker::GenApi::CLock, [576](#)
- m\_enabled
  - Spinnaker::GenICam::CGlobalLockUnlocker, [565](#)
- m\_lock
  - Spinnaker::GenApi::CLock, [576](#)
- m\_lockEx
  - Spinnaker::GenApi::CLockEx, [579](#)
- m\_pCameraBaseData
  - Spinnaker::ICameraBase, [672](#)
- m\_pCameraListData
  - Spinnaker::ICameraList, [674](#)
- m\_pChunkAdapter
  - Spinnaker::GenApi::CChunkAdapter, [531](#)
- m\_pChunkPort
  - Spinnaker::GenApi::CChunkPort, [542](#)
- m\_pEnumeration
  - Spinnaker::GenApi::EnumNode, [631](#)
- m\_pEventAdapter
  - Spinnaker::GenApi::CEventAdapter, [547](#)
- m\_pEventData
  - Spinnaker::Event, [637](#)
- m\_pEventPort
  - Spinnaker::GenApi::CEventPort, [557](#)
- m\_plImageData
  - Spinnaker::Image, [723](#)
- m\_plInterfaceData
  - Spinnaker::IInterface, [698](#)
- m\_plInterfaceListData
  - Spinnaker::IInterfaceList, [702](#)
- m\_pNode
  - Spinnaker::GenApi::CEventPort, [557](#)
  - Spinnaker::GenApi::CNodeCallback, [581](#)
- m\_pNodeData
  - Spinnaker::GenApi::Node, [777](#)
- m\_pNodeMap
  - Spinnaker::GenApi::Node, [778](#)
- m\_pPort
  - Spinnaker::GenApi::CChunkPort, [542](#)
- m\_pPortAdapter
  - Spinnaker::GenApi::CChunkPort, [542](#)
  - Spinnaker::GenApi::CEventPort, [557](#)
- m\_pWriteList
  - Spinnaker::GenApi::CPortWriteList, [607](#)
- m\_pT
  - Spinnaker::BasePtr, [382](#)
  - Spinnaker::GenApi::CPointer, [602](#)
- m\_ptrPort
  - Spinnaker::GenApi::CPortImpl, [604](#)

- MACAddress
  - Types Enums, [313](#)
- MJPGOption, [769](#)
  - Spinnaker::Video::MJPGOption, [770](#)
- Magic
  - GVCP\_REQUEST\_HEADER, [664](#)
- Major
  - Spinnaker::GenICam::Version\_t, [850](#)
- major
  - Spinnaker::LibraryVersion, [758](#)
- make\_NodeCallback
  - NodeCallback Class, [282](#)
- max\_size
  - Spinnaker::GenICam::gcstring, [656](#)
- MaxDeviceResetTime
  - Spinnaker::Camera, [491](#)
- MemSet
  - Spinnaker::GenApi::CTestPortStruct, [614](#)
- Member\_NodeCallback
  - Spinnaker::GenApi::Member\_NodeCallback, [769](#)
- Member\_NodeCallback < Client, Member >, [767](#)
- MergeXMLFiles
  - INodeMapDyn Interface, [264](#)
- Minor
  - Spinnaker::GenICam::Version\_t, [850](#)
- minor
  - Spinnaker::LibraryVersion, [758](#)
- NEAREST\_NEIGHBOR
  - Spinnaker Definitions, [161](#)
- NO\_COLOR\_PROCESSING
  - Spinnaker Definitions, [161](#)
- NO\_POLARIZATION
  - Spinnaker Definitions, [166](#)
- NONE
  - Spinnaker::TIFFOption, [824](#)
- NUM\_ACQUISITIONMODE
  - CameraDefs Class, [68](#)
- NUM\_ACQUISITIONSTATUSSELECTOR
  - CameraDefs Class, [69](#)
- NUM\_ACTIONUNCONDITIONALMODE
  - CameraDefs Class, [69](#)
- NUM\_ADCBITDEPTH
  - CameraDefs Class, [69](#)
- NUM\_AUTOALGORITHMSELECTOR
  - CameraDefs Class, [69](#)
- NUM\_AUTOEXPOSURECONTROLPRIORITY
  - CameraDefs Class, [70](#)
- NUM\_AUTOEXPOSURELIGHTINGMODE
  - CameraDefs Class, [70](#)
- NUM\_AUTOEXPOSUREMETERINGMODE
  - CameraDefs Class, [70](#)
- NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO
  - CameraDefs Class, [71](#)
- NUM\_BALANCERATIOSELECTOR
  - CameraDefs Class, [71](#)
- NUM\_BALANCEWHITEAUTOPROFILE
  - CameraDefs Class, [71](#)
- NUM\_BALANCEWHITEAUTO
  - CameraDefs Class, [71](#)
- NUM\_BINNINGHORIZONTALMODE
  - CameraDefs Class, [72](#)
- NUM\_BINNINGSELECTOR
  - CameraDefs Class, [72](#)
- NUM\_BINNINGVERTICALMODE
  - CameraDefs Class, [72](#)
- NUM\_BLACKLEVELAUTOBALANCE
  - CameraDefs Class, [72](#)
- NUM\_BLACKLEVELAUTO
  - CameraDefs Class, [73](#)
- NUM\_BLACKLEVELSELECTOR
  - CameraDefs Class, [73](#)
- NUM\_CHUNKBLACKLEVELSELECTOR
  - CameraDefs Class, [73](#)
- NUM\_CHUNKCOUNTERSELECTOR
  - CameraDefs Class, [73](#)
- NUM\_CHUNKENCODERSELECTOR
  - CameraDefs Class, [74](#)
- NUM\_CHUNKENCODERSTATUS
  - CameraDefs Class, [74](#)
- NUM\_CHUNKEXPOSURETIMESELECTOR
  - CameraDefs Class, [74](#)
- NUM\_CHUNKGAINSELECTOR
  - CameraDefs Class, [75](#)
- NUM\_CHUNKIMAGECOMPONENT
  - CameraDefs Class, [75](#)
- NUM\_CHUNKPIXELFORMAT
  - CameraDefs Class, [75](#)
- NUM\_CHUNKREGIONID
  - CameraDefs Class, [76](#)
- NUM\_CHUNKSCAN3DCOORDINATEREFERENCE↵
  - SELECTOR
  - CameraDefs Class, [76](#)
- NUM\_CHUNKSCAN3DCOORDINATESELECTOR
  - CameraDefs Class, [76](#)
- NUM\_CHUNKSCAN3DCOORDINATESYSTEMREF↵
  - ERENCE
  - CameraDefs Class, [77](#)
- NUM\_CHUNKSCAN3DCOORDINATESYSTEM
  - CameraDefs Class, [76](#)
- NUM\_CHUNKSCAN3DCOORDINATETRANSFORM↵
  - SELECTOR
  - CameraDefs Class, [77](#)
- NUM\_CHUNKSCAN3DDISTANCEUNIT
  - CameraDefs Class, [77](#)
- NUM\_CHUNKSCAN3DOUTPUTMODE
  - CameraDefs Class, [78](#)
- NUM\_CHUNKSELECTOR
  - CameraDefs Class, [79](#)
- NUM\_CHUNKSOURCEID
  - CameraDefs Class, [79](#)
- NUM\_CHUNKTIMERSELECTOR
  - CameraDefs Class, [79](#)
- NUM\_CHUNKTRANSFERSTREAMID
  - CameraDefs Class, [79](#)
- NUM\_CLCONFIGURATION
  - CameraDefs Class, [80](#)

- NUM\_CLTIMESLOTSCOUNT
  - CameraDefs Class, [80](#)
- NUM\_COLORTRANSFORMATIONSELECTOR
  - CameraDefs Class, [80](#)
- NUM\_COLORTRANSFORMATIONVALUESELECTOR
  - CameraDefs Class, [81](#)
- NUM\_COUNTEREVENTACTIVATION
  - CameraDefs Class, [81](#)
- NUM\_COUNTEREVENTSOURCE
  - CameraDefs Class, [82](#)
- NUM\_COUNTERRESETACTIVATION
  - CameraDefs Class, [82](#)
- NUM\_COUNTERRESETSOURCE
  - CameraDefs Class, [82](#)
- NUM\_COUNTERSELECTOR
  - CameraDefs Class, [83](#)
- NUM\_COUNTERSTATUS
  - CameraDefs Class, [83](#)
- NUM\_COUNTERTRIGGERACTIVATION
  - CameraDefs Class, [83](#)
- NUM\_COUNTERTRIGGERSOURCE
  - CameraDefs Class, [84](#)
- NUM\_CXPCONNECTIONTESTMODE
  - CameraDefs Class, [84](#)
- NUM\_CXPLINKCONFIGURATIONPREFERRED
  - CameraDefs Class, [86](#)
- NUM\_CXPLINKCONFIGURATIONSTATUS
  - CameraDefs Class, [87](#)
- NUM\_CXPLINKCONFIGURATION
  - CameraDefs Class, [85](#)
- NUM\_CXPPOCXPSTATUS
  - CameraDefs Class, [87](#)
- NUM\_DECIMATIONHORIZONTALMODE
  - CameraDefs Class, [87](#)
- NUM\_DECIMATIONSELECTOR
  - CameraDefs Class, [88](#)
- NUM\_DECIMATIONVERTICALMODE
  - CameraDefs Class, [88](#)
- NUM\_DEFECTCORRECTIONMODE
  - CameraDefs Class, [88](#)
- NUM\_DEINTERLACING
  - CameraDefs Class, [88](#)
- NUM\_DEVICECHARACTERSET
  - CameraDefs Class, [89](#)
- NUM\_DEVICECLOCKSELECTOR
  - CameraDefs Class, [89](#)
- NUM\_DEVICECONNECTIONSTATUS
  - CameraDefs Class, [89](#)
- NUM\_DEVICEINDICATORMODE
  - CameraDefs Class, [89](#)
- NUM\_DEVICELINKHEARTBEATMODE
  - CameraDefs Class, [90](#)
- NUM\_DEVICELINKTHROUGHPUTLIMITMODE
  - CameraDefs Class, [90](#)
- NUM\_DEVICEPOWERSUPPLYSELECTOR
  - CameraDefs Class, [90](#)
- NUM\_DEVICEREGISTERSENDIANNESSESS
  - CameraDefs Class, [90](#)
- NUM\_DEVICESCANTYPE
  - CameraDefs Class, [90](#)
- NUM\_DEVICESERIALPORTBAUDRATE
  - CameraDefs Class, [91](#)
- NUM\_DEVICESERIALPORTSELECTOR
  - CameraDefs Class, [91](#)
- NUM\_DEVICESTREAMCHANNELENDIANNESSESS
  - CameraDefs Class, [91](#)
- NUM\_DEVICESTREAMCHANNELTYPE
  - CameraDefs Class, [91](#)
- NUM\_DEVICETAPGEOMETRY
  - CameraDefs Class, [93](#)
- NUM\_DEVICETEMPERATURESELECTOR
  - CameraDefs Class, [93](#)
- NUM\_DEVICETLTYPE
  - CameraDefs Class, [93](#)
- NUM\_DEVICETYPE
  - CameraDefs Class, [93](#)
- NUM\_ENCODERMODE
  - CameraDefs Class, [94](#)
- NUM\_ENCODEROUTPUTMODE
  - CameraDefs Class, [94](#)
- NUM\_ENCODERRESETACTIVATION
  - CameraDefs Class, [94](#)
- NUM\_ENCODERRESETSOURCE
  - CameraDefs Class, [95](#)
- NUM\_ENCODERSELECTOR
  - CameraDefs Class, [96](#)
- NUM\_ENCODERSOURCEA
  - CameraDefs Class, [96](#)
- NUM\_ENCODERSOURCEB
  - CameraDefs Class, [96](#)
- NUM\_ENCODERSTATUS
  - CameraDefs Class, [96](#)
- NUM\_EVENTNOTIFICATION
  - CameraDefs Class, [97](#)
- NUM\_EVENTSELECTOR
  - CameraDefs Class, [97](#)
- NUM\_EXPOSUREACTIVEMODE
  - CameraDefs Class, [97](#)
- NUM\_EXPOSUREAUTO
  - CameraDefs Class, [97](#)
- NUM\_EXPOSUREMODE
  - CameraDefs Class, [98](#)
- NUM\_EXPOSURETIMEMODE
  - CameraDefs Class, [98](#)
- NUM\_EXPOSURETIMESELECTOR
  - CameraDefs Class, [98](#)
- NUM\_FILEOPENMODE
  - CameraDefs Class, [99](#)
- NUM\_FILEOPERATIONSELECTOR
  - CameraDefs Class, [99](#)
- NUM\_FILEOPERATIONSTATUS
  - CameraDefs Class, [99](#)
- NUM\_FILESELECTOR
  - CameraDefs Class, [99](#)
- NUM\_GAINAUTOBALANCE
  - CameraDefs Class, [100](#)

- NUM\_GAINAUTO
  - CameraDefs Class, [100](#)
- NUM\_GAINSELECTOR
  - CameraDefs Class, [100](#)
- NUM\_GEVCCP
  - CameraDefs Class, [100](#)
- NUM\_GEVCURRENTPHYSICALLINKCONFIGURATION↔
  - ION
  - CameraDefs Class, [101](#)
- NUM\_GEVGVCPEXTENDEDSTATUSCODESSELE↔
  - CTOR
  - CameraDefs Class, [101](#)
- NUM\_GEVGVSPEXTENDEDIDMODE
  - CameraDefs Class, [101](#)
- NUM\_GEVEEEE1588CLOCKACCURACY
  - CameraDefs Class, [101](#)
- NUM\_GEVEEEE1588MODE
  - CameraDefs Class, [102](#)
- NUM\_GEVEEEE1588STATUS
  - CameraDefs Class, [102](#)
- NUM\_GEVIPCONFIGURATIONSTATUS
  - CameraDefs Class, [102](#)
- NUM\_GEVPHYSCALLINKCONFIGURATION
  - CameraDefs Class, [102](#)
- NUM\_GEVSUPPORTEDOPTIONSELECTOR
  - CameraDefs Class, [103](#)
- NUM\_IMAGECOMPONENTSELECTOR
  - CameraDefs Class, [104](#)
- NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION
  - CameraDefs Class, [104](#)
- NUM\_IMAGECOMPRESSIONMODE
  - CameraDefs Class, [104](#)
- NUM\_IMAGECOMPRESSIONRATEOPTION
  - CameraDefs Class, [104](#)
- NUM\_LINEFORMAT
  - CameraDefs Class, [105](#)
- NUM\_LINEINPUTFILTERSELECTOR
  - CameraDefs Class, [105](#)
- NUM\_LINEMODE
  - CameraDefs Class, [105](#)
- NUM\_LINESELECTOR
  - CameraDefs Class, [105](#)
- NUM\_LINESOURCE
  - CameraDefs Class, [106](#)
- NUM\_LOGICBLOCKLUTINPUTACTIVATION
  - CameraDefs Class, [106](#)
- NUM\_LOGICBLOCKLUTINPUTSELECTOR
  - CameraDefs Class, [106](#)
- NUM\_LOGICBLOCKLUTINPUTSOURCE
  - CameraDefs Class, [107](#)
- NUM\_LOGICBLOCKLUTSELECTOR
  - CameraDefs Class, [107](#)
- NUM\_LOGICBLOCKSELECTOR
  - CameraDefs Class, [107](#)
- NUM\_LUTSELECTOR
  - CameraDefs Class, [108](#)
- NUM\_PIXELCOLORFILTER
  - CameraDefs Class, [108](#)
- NUM\_PIXELFORMATINFOSELECTOR
  - CameraDefs Class, [119](#)
- NUM\_PIXELFORMAT
  - CameraDefs Class, [113](#)
- NUM\_PIXELSIZE
  - CameraDefs Class, [119](#)
- NUM\_REGIONDESTINATION
  - CameraDefs Class, [119](#)
- NUM\_REGIONMODE
  - CameraDefs Class, [120](#)
- NUM\_REGIONSELECTOR
  - CameraDefs Class, [120](#)
- NUM\_RGBTRANSFORMLIGHTSOURCE
  - CameraDefs Class, [120](#)
- NUM\_SCAN3DCOORDINATEREFERENCESELECT↔
  - OR
  - CameraDefs Class, [121](#)
- NUM\_SCAN3DCOORDINATESELECTOR
  - CameraDefs Class, [121](#)
- NUM\_SCAN3DCOORDINATESYSTEMREFERENCE
  - CameraDefs Class, [121](#)
- NUM\_SCAN3DCOORDINATESYSTEM
  - CameraDefs Class, [121](#)
- NUM\_SCAN3DCOORDINATETRANSFORMSELEC↔
  - TOR
  - CameraDefs Class, [122](#)
- NUM\_SCAN3DDISTANCEUNIT
  - CameraDefs Class, [122](#)
- NUM\_SCAN3DOUTPUTMODE
  - CameraDefs Class, [123](#)
- NUM\_SENSORDIGITIZATIONTAPS
  - CameraDefs Class, [123](#)
- NUM\_SENSORSHUTTERMODE
  - CameraDefs Class, [123](#)
- NUM\_SENSORTAPS
  - CameraDefs Class, [123](#)
- NUM\_SEQUENCERCONFIGURATIONMODE
  - CameraDefs Class, [124](#)
- NUM\_SEQUENCERCONFIGURATIONVALID
  - CameraDefs Class, [124](#)
- NUM\_SEQUENCERMODE
  - CameraDefs Class, [124](#)
- NUM\_SEQUENCERSETVALID
  - CameraDefs Class, [124](#)
- NUM\_SEQUENCERTRIGGERACTIVATION
  - CameraDefs Class, [124](#)
- NUM\_SEQUENCERTRIGGERSOURCE
  - CameraDefs Class, [125](#)
- NUM\_SERIALPORTBAUDRATE
  - CameraDefs Class, [125](#)
- NUM\_SERIALPORTPARITY
  - CameraDefs Class, [125](#)
- NUM\_SERIALPORTSELECTOR
  - CameraDefs Class, [126](#)
- NUM\_SERIALPORTSOURCE
  - CameraDefs Class, [126](#)
- NUM\_SERIALPORTSTOPBITS
  - CameraDefs Class, [126](#)



- NUM\_SOFTWARESIGNALSELECTOR
  - CameraDefs Class, [126](#)
- NUM\_SOURCESELECTOR
  - CameraDefs Class, [127](#)
- NUM\_STATISTICS\_CHANNELS
  - Spinnaker Definitions, [167](#)
- NUM\_TESTPATTERNGENERATORSELECTOR
  - CameraDefs Class, [127](#)
- NUM\_TESTPATTERN
  - CameraDefs Class, [127](#)
- NUM\_TIMERSELECTOR
  - CameraDefs Class, [127](#)
- NUM\_TIMERSTATUS
  - CameraDefs Class, [128](#)
- NUM\_TIMERTRIGGERACTIVATION
  - CameraDefs Class, [128](#)
- NUM\_TIMERTRIGGERSOURCE
  - CameraDefs Class, [129](#)
- NUM\_TRANSFERCOMPONENTSELECTOR
  - CameraDefs Class, [130](#)
- NUM\_TRANSFERCONTROLMODE
  - CameraDefs Class, [130](#)
- NUM\_TRANSFEROPERATIONMODE
  - CameraDefs Class, [130](#)
- NUM\_TRANSFERQUEUEMODE
  - CameraDefs Class, [130](#)
- NUM\_TRANSFERSELECTOR
  - CameraDefs Class, [131](#)
- NUM\_TRANSFERSTATUSSELECTOR
  - CameraDefs Class, [131](#)
- NUM\_TRANSFERTRIGGERACTIVATION
  - CameraDefs Class, [131](#)
- NUM\_TRANSFERTRIGGERMODE
  - CameraDefs Class, [132](#)
- NUM\_TRANSFERTRIGGERSELECTOR
  - CameraDefs Class, [132](#)
- NUM\_TRANSFERTRIGGERSOURCE
  - CameraDefs Class, [133](#)
- NUM\_TRIGGERACTIVATION
  - CameraDefs Class, [133](#)
- NUM\_TRIGGERMODE
  - CameraDefs Class, [133](#)
- NUM\_TRIGGEROVERLAP
  - CameraDefs Class, [134](#)
- NUM\_TRIGGERSELECTOR
  - CameraDefs Class, [134](#)
- NUM\_TRIGGERSOURCE
  - CameraDefs Class, [134](#)
- NUM\_USEROUTPUTSELECTOR
  - CameraDefs Class, [135](#)
- NUM\_USERSETDEFAULT
  - CameraDefs Class, [135](#)
- NUM\_USERSETSELECTOR
  - CameraDefs Class, [135](#)
- NUM\_WHITECLIPSELECTOR
  - CameraDefs Class, [135](#)
- NUMDEVICEACCESSSTATUS
  - TransportLayerDefs Class, [175](#)
- NUMDEVICECURRENTSPEED
  - TransportLayerDefs Class, [176](#)
- NUMDEVICEENDIANESSMECHANISM
  - TransportLayerDefs Class, [176](#)
- NUMDEVICETYPE
  - TransportLayerDefs Class, [176](#)
- NUMGENICAMXMLLOCATION
  - TransportLayerDefs Class, [177](#)
- NUMGEVCCP
  - TransportLayerDefs Class, [177](#)
- NUMGUIXMLLOCATION
  - TransportLayerDefs Class, [177](#)
- NUMPOESTATUS
  - TransportLayerDefs Class, [177](#)
- NUMSTREAMBUFFERCOUNTMODE
  - TransportLayerDefs Class, [178](#)
- NUMSTREAMBUFFERHANDLINGMODE
  - TransportLayerDefs Class, [178](#)
- NUMSTREAMDEFAULTBUFFERCOUNTMODE
  - TransportLayerDefs Class, [178](#)
- NUMSTREAMTYPE
  - TransportLayerDefs Class, [179](#)
- NA
  - Types Enums, [311](#)
- NI
  - Types Enums, [311](#)
- No
  - Types Enums, [315](#)
- NoCache
  - Types Enums, [311](#)
- noIncrement
  - Types Enums, [312](#)
- Node, [770](#)
  - Spinnaker::GenApi::Node, [773](#)
- Node Class, [280](#)
- NodeCallback Class, [281](#)
  - cbPostInsideLock, [282](#)
  - cbPostOutsideLock, [282](#)
  - Deregister, [282](#)
  - ECallbackType, [282](#)
  - make\_NodeCallback, [282](#)
  - Register, [282](#)
- NodeList\_t
  - Spinnaker GenApi Interfaces, [201](#)
- NodeMap, [778](#)
  - Spinnaker::GenApi::CLock, [576](#)
  - Spinnaker::GenApi::NodeMap, [780](#)
- NodeMap Class, [283](#)
- NodeMapFactory Class, [284](#)
  - CacheUsage\_Automatic, [284](#)
  - CacheUsage\_ForceRead, [284](#)
  - CacheUsage\_ForceWrite, [284](#)
  - CacheUsage\_Ignore, [284](#)
  - ContentType\_Xml, [285](#)
  - ContentType\_ZippedXml, [285](#)
  - ECacheUsage\_t, [284](#)
  - EContentType\_t, [284](#)
- NodeMapRef Class, [286](#)

- None
  - Types Enums, [314](#)
- npos
  - Spinnaker::GenICam::gcstring, [657](#)
- NumAttachedChunks
  - AttachStatistics\_t, [378](#)
- NumChunkPorts
  - AttachStatistics\_t, [378](#)
- NumChunks
  - AttachStatistics\_t, [378](#)
- NumLinks
  - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics\_t, [784](#)
- NumNodes
  - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics\_t, [784](#)
- NumProperties
  - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics\_t, [784](#)
- NumStrings
  - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics\_t, [784](#)
- ODevFileStream
  - Spinnaker::GenApi, [371](#)
- ODevFileStreamBase< CharType, Traits >, [785](#)
- ODevFileStreamBuf
  - Spinnaker::GenApi::ODevFileStreamBuf, [788](#)
- ODevFileStreamBuf< CharType, Traits >, [787](#)
- OffsetX
  - Spinnaker::Camera, [491](#)
- OffsetY
  - Spinnaker::Camera, [491](#)
- OnDeviceArrival
  - Spinnaker::ArrivalEvent, [377](#)
  - Spinnaker::IArrivalEvent, [667](#)
  - Spinnaker::IInterfaceEvent, [700](#)
  - Spinnaker::InterfaceEvent, [745](#)
- OnDeviceEvent
  - Spinnaker::DeviceEvent, [617](#)
  - Spinnaker::IDeviceEvent, [684](#)
- OnDeviceRemoval
  - Spinnaker::IInterfaceEvent, [700](#)
  - Spinnaker::IRemovalEvent, [753](#)
  - Spinnaker::InterfaceEvent, [745](#)
  - Spinnaker::RemovalEvent, [804](#)
- OnImageEvent
  - Spinnaker::IImageEvent, [693](#)
  - Spinnaker::ImageEvent, [725](#)
- OnLogEvent
  - Spinnaker::ILoggingEvent, [703](#)
  - Spinnaker::LoggingEvent, [763](#)
- Open
  - Spinnaker::Video::SpinVideo, [808](#)
- open
  - Spinnaker::GenApi::IDevFileStreamBase, [681](#)
  - Spinnaker::GenApi::IDevFileStreamBuf, [683](#)
  - Spinnaker::GenApi::ODevFileStreamBase, [786](#)
  - Spinnaker::GenApi::ODevFileStreamBuf, [788](#)
- openFile
  - Spinnaker::GenApi::FileProtocolAdapter, [644](#)
- operator bool
  - Spinnaker::BasePtr, [381](#)
  - Spinnaker::GenApi::CPointer, [600](#)
- operator const char \*
  - Spinnaker::GenICam::gcstring, [656](#)
- operator delete
  - Spinnaker::GenApi::double\_autovector\_t, [618](#)
  - Spinnaker::GenApi::int64\_autovector\_t, [734](#)
  - Spinnaker::GenICam::gcstring, [656](#)
- operator new
  - Spinnaker::GenApi::double\_autovector\_t, [618](#)
  - Spinnaker::GenApi::int64\_autovector\_t, [734](#)
  - Spinnaker::GenICam::gcstring, [656](#)
- operator T \*
  - Spinnaker::BasePtr, [381](#)
  - Spinnaker::GenApi::CPointer, [600](#)
- operator unsigned int
  - Spinnaker::GenApi::Counter, [598](#)
- operator!=
  - INode Interface, [259](#)
  - Spinnaker::Exception, [641](#)
  - Spinnaker::GenApi::CPointer, [600](#), [601](#)
  - Spinnaker::GenApi::Node, [777](#)
  - Spinnaker::GenICam::gcstring, [656](#)
- operator<
  - Spinnaker::GenICam::gcstring, [656](#)
- operator<<
  - GCString.h, [997](#)
  - Spinnaker GenApi Classes, [197](#)
- operator>
  - Spinnaker::GenICam::gcstring, [657](#)
- operator>>
  - GCString.h, [997](#)
  - Spinnaker GenApi Classes, [198](#)
- operator\*
  - IEnumeration Interface, [246](#)
  - Spinnaker::GenApi::CPointer, [601](#)
  - Spinnaker::GenApi::EnumNode, [631](#)
  - Spinnaker::GenApi::FloatNode, [649](#)
  - Spinnaker::GenApi::IntegerNode, [738](#)
  - Spinnaker::GenApi::StringNode, [812](#)
- operator()
  - IBoolean Interface, [235](#)
  - Spinnaker::GenApi::CEnumerationTRef, [545](#)
  - Spinnaker::GenApi::CNodeCallback, [581](#)
  - Spinnaker::GenApi::CPointer, [601](#)
  - Spinnaker::GenApi::CommandNode, [597](#)
  - Spinnaker::GenApi::FloatNode, [649](#)
  - Spinnaker::GenApi::Function\_NodeCallback, [653](#)
  - Spinnaker::GenApi::IntegerNode, [738](#)
  - Spinnaker::GenApi::Member\_NodeCallback, [769](#)
  - Spinnaker::GenApi::StringNode, [812](#)
- operator+
  - Spinnaker::GenICam::gcstring, [657](#)
- operator++
  - Spinnaker::GenApi::Counter, [598](#)



- operator+=
  - Spinnaker::GenICam::gcstring, [656](#)
- operator->
  - Spinnaker::BasePtr, [381](#)
  - Spinnaker::GenApi::CPointer, [601](#)
- operator--
  - Spinnaker::GenApi::Counter, [598](#)
- operator=
  - CameraPtr Class, [137](#)
  - IBoolean Interface, [236](#)
  - IEnumerationT Interface, [247](#)
  - IFloat Interface, [251](#)
  - Integer Interface, [252](#)
  - Spinnaker GenApi Classes, [197](#)
  - Spinnaker::ArrivalEvent, [377](#)
  - Spinnaker::BasePtr, [381](#)
  - Spinnaker::CameraBase, [520](#)
  - Spinnaker::CameraList, [525](#)
  - Spinnaker::DeviceEvent, [617](#)
  - Spinnaker::Event, [637](#)
  - Spinnaker::Exception, [641](#)
  - Spinnaker::GenApi::BooleanNode, [385](#)
  - Spinnaker::GenApi::CEnumerationTRef, [545](#)
  - Spinnaker::GenApi::CFloatPtr, [561](#)
  - Spinnaker::GenApi::CNodeMapFactory, [588](#)
  - Spinnaker::GenApi::CNodeMapRef, [591](#)
  - Spinnaker::GenApi::CPointer, [601](#)
  - Spinnaker::GenApi::EnumNode, [631](#)
  - Spinnaker::GenApi::FloatNode, [649](#)
  - Spinnaker::GenApi::IntegerNode, [739](#)
  - Spinnaker::GenApi::StringNode, [812](#)
  - Spinnaker::GenApi::double\_autovector\_t, [618](#)
  - Spinnaker::GenApi::int64\_autovector\_t, [734](#)
  - Spinnaker::GenICam::gcstring, [656](#)
  - Spinnaker::IArrivalEvent, [667](#)
  - Spinnaker::ICameraBase, [671](#)
  - Spinnaker::ICameraList, [674](#)
  - Spinnaker::IDeviceEvent, [685](#)
  - Spinnaker::IImageEvent, [693](#)
  - Spinnaker::IInterface, [698](#)
  - Spinnaker::IInterfaceEvent, [700](#)
  - Spinnaker::IInterfaceList, [701](#)
  - Spinnaker::ILoggingEvent, [703](#)
  - Spinnaker::IRemovalEvent, [753](#)
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::ImageEvent, [725](#)
  - Spinnaker::ImagePtr, [727](#)
  - Spinnaker::ImageStatistics, [732](#)
  - Spinnaker::InterfaceEvent, [745](#)
  - Spinnaker::InterfaceList, [747](#)
  - Spinnaker::InterfacePtr, [749](#)
  - Spinnaker::LoggingEvent, [763](#)
  - Spinnaker::LoggingEventDataPtr, [767](#)
  - Spinnaker::RemovalEvent, [804](#)
- operator==
  - INode Interface, [259](#)
  - Spinnaker::BasePtr, [381](#), [382](#)
  - Spinnaker::Exception, [641](#)
  - Spinnaker::GenApi::CFeatureBag, [559](#)
  - Spinnaker::GenApi::CPointer, [601](#)
  - Spinnaker::GenApi::Node, [777](#)
  - Spinnaker::GenICam::gcstring, [656](#), [657](#)
- operator[]
  - Spinnaker::CameraList, [525](#)
  - Spinnaker::GenApi::double\_autovector\_t, [618](#)
  - Spinnaker::GenApi::int64\_autovector\_t, [734](#)
  - Spinnaker::ICameraList, [674](#)
  - Spinnaker::IInterfaceList, [702](#)
  - Spinnaker::InterfaceList, [748](#)
- ostream\_type
  - Spinnaker::GenApi::ODevFileStreamBase, [786](#)
- overflow
  - Spinnaker::GenApi::ODevFileStreamBuf, [788](#)
- PACKBITS
  - Spinnaker::TIFFOption, [824](#)
- PAYLOAD\_TYPE\_CHUNK\_DATA
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_CHUNK\_ONLY
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_CUSTOM\_ID
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_DEVICE\_SPECIFIC
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_EXTENDED\_CHUNK
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_FILE
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_H264
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_IMAGE
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_JPEG2000
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_JPEG
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_MULTI\_PART
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_RAW\_DATA
  - Spinnaker Definitions, [164](#)
- PAYLOAD\_TYPE\_UNKNOWN
  - Spinnaker Definitions, [164](#)
- PGMOption, [788](#)
  - Spinnaker::PGMOption, [789](#)
- PGM
  - Spinnaker Definitions, [163](#)
- PMEBERFUNC
  - Spinnaker::GenApi::Member\_NodeCallback, [768](#)
- PNGOption, [789](#)
  - Spinnaker::PNGOption, [790](#)
- PNG
  - Spinnaker Definitions, [163](#)
- POEStatus
  - Spinnaker::TransportLayerInterface, [839](#)
- POEStatus\_NotSupported
  - TransportLayerDefs Class, [177](#)
- POEStatus\_PowerOff

- TransportLayerDefs Class, [177](#)
- POEStatus\_PowerOn
  - TransportLayerDefs Class, [177](#)
- POEStatusEnum
  - TransportLayerDefs Class, [177](#)
- PPMOption, [799](#)
  - Spinnaker::PPMOption, [799](#)
- PPM
  - Spinnaker Definitions, [163](#)
- PacketResendRequestCount
  - Spinnaker::Camera, [491](#)
- PayloadSize
  - Spinnaker::Camera, [491](#)
- PayloadTypeInfoIds
  - Spinnaker Definitions, [164](#)
- pbackfail
  - Spinnaker::GenApi::IDevFileStreamBuf, [683](#)
- PersistFeature
  - Spinnaker::GenApi, [371](#)
  - Spinnaker::GenApi::CFeatureBag, [559](#)
- Persistence Class, [287](#)
- PixelColorFilter
  - Spinnaker::Camera, [491](#)
- PixelColorFilter\_BayerBG
  - CameraDefs Class, [108](#)
- PixelColorFilter\_BayerGB
  - CameraDefs Class, [108](#)
- PixelColorFilter\_BayerGR
  - CameraDefs Class, [108](#)
- PixelColorFilter\_BayerRG
  - CameraDefs Class, [108](#)
- PixelColorFilter\_None
  - CameraDefs Class, [108](#)
- PixelColorFilterEnums
  - CameraDefs Class, [108](#)
- PixelDynamicRangeMax
  - Spinnaker::Camera, [492](#)
- PixelDynamicRangeMin
  - Spinnaker::Camera, [492](#)
- PixelFormat
  - Spinnaker::Camera, [492](#)
- PixelFormat\_B10
  - CameraDefs Class, [110](#)
- PixelFormat\_B12
  - CameraDefs Class, [110](#)
- PixelFormat\_B12\_Jpeg
  - CameraDefs Class, [113](#)
- PixelFormat\_B16
  - CameraDefs Class, [110](#)
- PixelFormat\_B8
  - CameraDefs Class, [110](#)
- PixelFormat\_BGR10
  - CameraDefs Class, [110](#)
- PixelFormat\_BGR10p
  - CameraDefs Class, [110](#)
- PixelFormat\_BGR12
  - CameraDefs Class, [110](#)
- PixelFormat\_BGR12p
  - CameraDefs Class, [110](#)
- PixelFormat\_BGR14
  - CameraDefs Class, [110](#)
- PixelFormat\_BGR16
  - CameraDefs Class, [110](#)
- PixelFormat\_BGR565p
  - CameraDefs Class, [110](#)
- PixelFormat\_BGR8
  - CameraDefs Class, [109](#)
- PixelFormat\_BGRa10
  - CameraDefs Class, [110](#)
- PixelFormat\_BGRa10p
  - CameraDefs Class, [110](#)
- PixelFormat\_BGRa12
  - CameraDefs Class, [110](#)
- PixelFormat\_BGRa12p
  - CameraDefs Class, [110](#)
- PixelFormat\_BGRa14
  - CameraDefs Class, [110](#)
- PixelFormat\_BGRa16
  - CameraDefs Class, [110](#)
- PixelFormat\_BGRa8
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerBG10
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerBG10Packed
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerBG10p
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerBG12
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerBG12Packed
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerBG12p
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerBG16
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerBG8
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerGB10
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGB10Packed
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGB10p
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGB12
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGB12Packed
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerGB12p
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGB16
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerGB8
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerGR10
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGR10Packed
  - CameraDefs Class, [109](#)

- CameraDefs Class, [109](#)
- PixelFormat\_BayerGR10p
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGR12
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGR12Packed
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerGR12p
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerGR16
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerGR8
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerRG10
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerRG10Packed
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerRG10p
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerRG12
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerRG12Packed
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerRG12p
  - CameraDefs Class, [109](#)
- PixelFormat\_BayerRG16
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerRG8
  - CameraDefs Class, [108](#)
- PixelFormat\_BayerRGPolarized10p
  - CameraDefs Class, [113](#)
- PixelFormat\_BayerRGPolarized12p
  - CameraDefs Class, [113](#)
- PixelFormat\_BayerRGPolarized16
  - CameraDefs Class, [113](#)
- PixelFormat\_BayerRGPolarized8
  - CameraDefs Class, [113](#)
- PixelFormat\_BiColorBGRG10
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorBGRG10p
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorBGRG12
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorBGRG12p
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorBGRG8
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorRGBG10
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorRGBG10p
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorRGBG12
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorRGBG12p
  - CameraDefs Class, [111](#)
- PixelFormat\_BiColorRGBG8
  - CameraDefs Class, [111](#)
- PixelFormat\_Confidence1
  - CameraDefs Class, [111](#)
- PixelFormat\_Confidence16
  - CameraDefs Class, [111](#)
- PixelFormat\_Confidence1p
  - CameraDefs Class, [111](#)
- PixelFormat\_Confidence32f
  - CameraDefs Class, [111](#)
- PixelFormat\_Confidence8
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_A10p
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_A12p
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_A16
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_A32f
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_A8
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_ABC10p
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_ABC10p\_Planar
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_ABC12p
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_ABC12p\_Planar
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_ABC16
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_ABC16\_Planar
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_ABC32f
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_ABC32f\_Planar
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_ABC8
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_ABC8\_Planar
  - CameraDefs Class, [110](#)
- PixelFormat\_Coord3D\_AC10p
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC10p\_Planar
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC12p
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC12p\_Planar
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC16
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC16\_Planar
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC32f
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC32f\_Planar
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC8
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_AC8\_Planar
  - CameraDefs Class, [111](#)

- CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_B10p
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_B12p
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_B16
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_B32f
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_B8
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_C10p
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_C12p
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_C16
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_C32f
  - CameraDefs Class, [111](#)
- PixelFormat\_Coord3D\_C8
  - CameraDefs Class, [111](#)
- PixelFormat\_G10
  - CameraDefs Class, [110](#)
- PixelFormat\_G12
  - CameraDefs Class, [110](#)
- PixelFormat\_G16
  - CameraDefs Class, [110](#)
- PixelFormat\_G8
  - CameraDefs Class, [110](#)
- PixelFormat\_GB12\_Jpeg
  - CameraDefs Class, [113](#)
- PixelFormat\_GR12\_Jpeg
  - CameraDefs Class, [113](#)
- PixelFormat\_Mono10
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono10Packed
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono10p
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono12
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono12Packed
  - CameraDefs Class, [108](#)
- PixelFormat\_Mono12p
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono14
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono16
  - CameraDefs Class, [108](#)
- PixelFormat\_Mono1p
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono2p
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono4p
  - CameraDefs Class, [109](#)
- PixelFormat\_Mono8
  - CameraDefs Class, [108](#)
- PixelFormat\_Mono8s
  - CameraDefs Class, [109](#)
- PixelFormat\_Polarized10p
  - CameraDefs Class, [113](#)
- PixelFormat\_Polarized12p
  - CameraDefs Class, [113](#)
- PixelFormat\_Polarized16
  - CameraDefs Class, [113](#)
- PixelFormat\_Polarized8
  - CameraDefs Class, [113](#)
- PixelFormat\_R10
  - CameraDefs Class, [110](#)
- PixelFormat\_R12
  - CameraDefs Class, [110](#)
- PixelFormat\_R12\_Jpeg
  - CameraDefs Class, [113](#)
- PixelFormat\_R16
  - CameraDefs Class, [110](#)
- PixelFormat\_R8
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB10
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB10\_Planar
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB10p
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB10p32
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB12
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB12\_Planar
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB12p
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB14
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB16
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB16\_Planar
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB565p
  - CameraDefs Class, [110](#)
- PixelFormat\_RGB8
  - CameraDefs Class, [109](#)
- PixelFormat\_RGB8\_Planar
  - CameraDefs Class, [109](#)
- PixelFormat\_RGB8Packed
  - CameraDefs Class, [108](#)
- PixelFormat\_RGBa10
  - CameraDefs Class, [109](#)
- PixelFormat\_RGBa10p
  - CameraDefs Class, [109](#)
- PixelFormat\_RGBa12
  - CameraDefs Class, [109](#)
- PixelFormat\_RGBa12p
  - CameraDefs Class, [109](#)
- PixelFormat\_RGBa14
  - CameraDefs Class, [109](#)
- PixelFormat\_RGBa16
  - CameraDefs Class, [109](#)

CameraDefs Class, [109](#)  
PixelFormat\_RGBA8  
    CameraDefs Class, [109](#)  
PixelFormat\_Raw16  
    CameraDefs Class, [113](#)  
PixelFormat\_Raw8  
    CameraDefs Class, [113](#)  
PixelFormat\_SCF1WBWG10  
    CameraDefs Class, [111](#)  
PixelFormat\_SCF1WBWG10p  
    CameraDefs Class, [111](#)  
PixelFormat\_SCF1WBWG12  
    CameraDefs Class, [111](#)  
PixelFormat\_SCF1WBWG12p  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WBWG14  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WBWG16  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WBWG8  
    CameraDefs Class, [111](#)  
PixelFormat\_SCF1WGWB10  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWB10p  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWB12  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWB12p  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWB14  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWB16  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWB8  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWR10  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWR10p  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWR12  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWR12p  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWR14  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWR16  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WGWR8  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WRWG10  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WRWG10p  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WRWG12  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WRWG12p  
    CameraDefs Class, [112](#)  
PixelFormat\_SCF1WRWG14

CameraDefs Class, [112](#)  
PixelFormat\_SCF1WRWG16  
CameraDefs Class, [112](#)  
PixelFormat\_SCF1WRWG8  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr10\_CbYCr  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr10p\_CbYCr  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr12\_CbYCr  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr12p\_CbYCr  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr411\_8  
CameraDefs Class, [109](#)  
PixelFormat\_YCbCr411\_8\_CbYYCrYY  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_10  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_10\_CbYCrY  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_10p  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_10p\_CbYCrY  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_12  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_12\_CbYCrY  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_12p  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_12p\_CbYCrY  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr422\_8  
CameraDefs Class, [109](#)  
PixelFormat\_YCbCr422\_8\_CbYCrY  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr601\_10\_CbYCr  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr601\_10p\_CbYCr  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr601\_12\_CbYCr  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr601\_12p\_CbYCr  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr601\_411\_8\_CbYYCrYY  
CameraDefs Class, [112](#)  
PixelFormat\_YCbCr601\_422\_10  
CameraDefs Class, [113](#)  
PixelFormat\_YCbCr601\_422\_10\_CbYCrY  
CameraDefs Class, [113](#)  
PixelFormat\_YCbCr601\_422\_10p  
CameraDefs Class, [113](#)  
PixelFormat\_YCbCr601\_422\_10p\_CbYCrY  
CameraDefs Class, [113](#)  
PixelFormat\_YCbCr601\_422\_12  
CameraDefs Class, [113](#)  
PixelFormat\_YCbCr601\_422\_12\_CbYCrY

- CameraDefs Class, [113](#)
- PixelFormat\_YCbCr601\_422\_12p
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr601\_422\_12p\_CbYCrY
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr601\_422\_8
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr601\_422\_8\_CbYCrY
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr601\_8\_CbYCr
  - CameraDefs Class, [112](#)
- PixelFormat\_YCbCr709\_10\_CbYCr
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_10p\_CbYCr
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_12\_CbYCr
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_12p\_CbYCr
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_411\_8\_CbYYCrYY
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_10
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_10\_CbYCrY
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_10p
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_10p\_CbYCrY
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_12
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_12\_CbYCrY
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_12p
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_12p\_CbYCrY
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_8
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_422\_8\_CbYCrY
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr709\_8\_CbYCr
  - CameraDefs Class, [113](#)
- PixelFormat\_YCbCr8
  - CameraDefs Class, [109](#)
- PixelFormat\_YCbCr8\_CbYCr
  - CameraDefs Class, [112](#)
- PixelFormat\_YUV411\_8\_UYYVYY
  - CameraDefs Class, [113](#)
- PixelFormat\_YUV411Packed
  - CameraDefs Class, [108](#)
- PixelFormat\_YUV422\_8
  - CameraDefs Class, [113](#)
- PixelFormat\_YUV422\_8\_UYVY
  - CameraDefs Class, [113](#)
- PixelFormat\_YUV422Packed
  - CameraDefs Class, [108](#)
- PixelFormat\_YUV444Packed
  - CameraDefs Class, [109](#)
- PixelFormat\_YUV8\_UYV
  - CameraDefs Class, [113](#)
- PixelFormatEnums
  - CameraDefs Class, [108](#)
- PixelFormatInfoD
  - Spinnaker::Camera, [492](#)
- PixelFormatInfoSelector
  - Spinnaker::Camera, [492](#)
- PixelFormatInfoSelector\_B10
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_B12
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_B16
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_B8
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGR10
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGR10p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGR12
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGR12p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGR14
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGR16
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGR565p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGR8
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGRa10
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGRa10p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGRa12
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGRa12p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGRa14
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGRa16
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BGRa8
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_BayerBG10
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerBG10p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerBG12
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerBG12p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerBG16
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerBG8
  - CameraDefs Class, [114](#)



- CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGB10
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGB10p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGB12
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGB12p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGB16
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGB8
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGR10
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGR10p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGR12
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGR12p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGR16
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerGR8
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerRG10
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerRG10p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerRG12
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerRG12p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerRG16
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerRG8
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_BayerRGPolarized10p
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_BayerRGPolarized12p
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_BayerRGPolarized16
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_BayerRGPolarized8
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_BiColorBGRG10
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_BiColorBGRG10p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_BiColorBGRG12
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_BiColorBGRG12p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_BiColorBGRG8
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_BiColorRGBG10
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_BiColorRGBG10p
  - CameraDefs Class, [116](#)
- CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_BiColorRGBG12
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_BiColorRGBG12p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_BiColorRGBG8
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Confidence1
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Confidence16
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Confidence1p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Confidence32f
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Confidence8
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_A10p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_A12p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_A16
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_A32f
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_A8
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_ABC10p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_Coord3D\_ABC10p\_Planar
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_Coord3D\_ABC12p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_ABC12p\_Planar
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_ABC16
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_ABC16\_Planar
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_ABC32f
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_ABC32f\_Planar
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_ABC8
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_Coord3D\_ABC8\_Planar
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_Coord3D\_AC10p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC10p\_Planar
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC12p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC12p\_Planar
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC16
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC16\_Planar

- CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC32f
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC32f\_Planar
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC8
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_AC8\_Planar
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_B10p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_B12p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_B16
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_B32f
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_B8
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_C10p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_C12p
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_C16
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_C32f
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_Coord3D\_C8
  - CameraDefs Class, [116](#)
- PixelFormatInfoSelector\_G10
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_G12
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_G16
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_G8
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_Mono10
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono10p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono12
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono12p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono14
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono16
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono1p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono2p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono4p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono8
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Mono8s
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_Polarized10p
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_Polarized12p
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_Polarized16
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_Polarized8
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_R10
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_R12
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_R16
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_R8
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB10
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB10\_Planar
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB10p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB10p32
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB12
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB12\_Planar
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB12p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB14
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB16
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB16\_Planar
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB565p
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB8
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGB8\_Planar
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGBa10
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_RGBa10p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_RGBa12
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_RGBa12p
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_RGBa14
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_RGBa16
  - CameraDefs Class, [115](#)
- PixelFormatInfoSelector\_RGBa8
  - CameraDefs Class, [114](#)
- PixelFormatInfoSelector\_SCF1WBWG10



- CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG10p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG12
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG12p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG14
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG16
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WBWG8
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB10
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB10p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB12
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB12p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB14
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB16
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWB8
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR10
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR10p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR12
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR12p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR14
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR16
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WGWR8
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG10
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG10p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG12
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG12p
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG14
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG16
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_SCF1WRWG8
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_YCbCr10\_CbYCr
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_YCbCr10p\_CbYCr
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_YCbCr12\_CbYCr
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_YCbCr12p\_CbYCr
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_YCbCr411\_8
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr411\_8\_CbYYCrYY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_10
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_10\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_10p
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_10p\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_12
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_12\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_12p
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_12p\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_8
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr422\_8\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_10\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_10p\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_12\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_12p\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_411\_8\_CbYYCrYY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_10
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_10\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_10p
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_10p\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_12
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_12\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_12p
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_12p\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_8
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_422\_8\_CbYCrY
  - CameraDefs Class, [118](#)

- CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr601\_8\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_10\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_10p\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_12\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_12p\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_411\_8\_CbYYCrYY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_10
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_10\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_10p
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_10p\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_12
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_12\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_12p
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_12p\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_8
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_422\_8\_CbYCrY
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr709\_8\_CbYCr
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelector\_YCbCr8
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_YCbCr8\_CbYCr
  - CameraDefs Class, [117](#)
- PixelFormatInfoSelector\_YUV411\_8\_UYYVYY
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_YUV422\_8
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_YUV422\_8\_UYVY
  - CameraDefs Class, [119](#)
- PixelFormatInfoSelector\_YUV8\_UYV
  - CameraDefs Class, [118](#)
- PixelFormatInfoSelectorEnums
  - CameraDefs Class, [113](#)
- PixelFormatIntType
  - Spinnaker Definitions, [164](#)
- PixelFormatNamespaceID
  - Spinnaker Definitions, [165](#)
- PixelSize
  - Spinnaker::Camera, [492](#)
- PixelSize\_Bpp1
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp10
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp12
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp14
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp16
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp2
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp20
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp24
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp30
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp32
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp36
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp4
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp48
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp64
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp8
  - CameraDefs Class, [119](#)
- PixelSize\_Bpp96
  - CameraDefs Class, [119](#)
- PixelSizeEnums
  - CameraDefs Class, [119](#)
- Pointer Class, [288](#)
  - CBasePtr, [289](#)
  - CBooleanPtr, [289](#)
  - CCategoryPtr, [289](#)
  - CChunkPortPtr, [290](#)
  - CCommandPtr, [290](#)
  - CDeviceInfoPtr, [290](#)
  - CEnumEntryPtr, [290](#)
  - CEnumerationPtr, [290](#)
  - CIntegerPtr, [290](#)
  - CNodeMapDynPtr, [290](#)
  - CNodeMapPtr, [290](#)
  - CNodePtr, [290](#)
  - CPortConstructPtr, [290](#)
  - CPortPtr, [291](#)
  - CPortRecorderPtr, [291](#)
  - CPortReplayPtr, [291](#)
  - CPortWriteListPtr, [291](#)
  - CRegisterPtr, [291](#)
  - CSelectorPtr, [291](#)
  - CStringPtr, [291](#)
  - CValuePtr, [291](#)
  - GetInterfaceName, [291](#)
  - IsAvailable, [291](#)
  - IsImplemented, [292](#)
  - IsReadable, [292](#)
  - IsWritable, [292](#)

- PolarizationAlgorithm
  - Spinnaker Definitions, [165](#)
- PolarizationResolution
  - Spinnaker Definitions, [166](#)
- Poll
  - INodeMap Interface, [261](#)
  - Spinnaker::GenApi::NodeMap, [783](#)
- PortImpl Class, [293](#)
- PortNode, [790](#)
  - Spinnaker::GenApi::PortNode, [792](#)
- PortNode Class, [294](#)
  - CPortRef, [294](#)
- PortRecorder, [794](#)
  - Spinnaker::GenApi::PortRecorder, [796](#)
- PortRecorder Class, [295](#)
  - CPortRecorderRef, [295](#)
- PortReplay, [797](#)
  - Spinnaker::GenApi::PortReplay, [798](#)
- PortReplay Class, [296](#)
- PortWriteList Class, [297](#)
- PowerSupplyCurrent
  - Spinnaker::Camera, [492](#)
- PowerSupplyVoltage
  - Spinnaker::Camera, [493](#)
- Prefix
  - U3V\_COMMAND\_HEADER, [845](#)
- Preprocess
  - Spinnaker::GenApi::CNodeMapFactory, [588](#)
- PreprocessXMLFromFile
  - INodeMapDyn Interface, [264](#)
- PreprocessXMLFromZIPFile
  - INodeMapDyn Interface, [265](#)
- progressive
  - Spinnaker::JPEGOption, [756](#)
- PureNumber
  - Types Enums, [313](#)
- QUADRANT\_I0\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- QUADRANT\_I135\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- QUADRANT\_I45\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- QUADRANT\_I90\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- QUARTER\_RESOLUTION
  - Spinnaker Definitions, [166](#)
- quality
  - Spinnaker::JPEGOption, [756](#)
  - Spinnaker::JPG2Option, [757](#)
  - Spinnaker::Video::MJPGOption, [770](#)
- RAW
  - Spinnaker Definitions, [163](#)
- RED
  - Spinnaker Definitions, [167](#)
- RIGOROUS
  - Spinnaker Definitions, [161](#)
- rdbuf
  - Spinnaker::GenApi::IDevFileStreamBase, [681](#)
  - Spinnaker::GenApi::ODevFileStreamBase, [786](#)
- Read
  - Spinnaker::GenApi::CChunkPort, [541](#)
  - Spinnaker::GenApi::CEventPort, [557](#)
  - Spinnaker::GenApi::CPortImpl, [604](#)
  - Spinnaker::GenApi::CRegisterPortImpl, [608](#)
  - Spinnaker::GenApi::CTestPortStruct, [614](#)
  - Spinnaker::GenApi::PortNode, [792](#)
- read
  - Spinnaker::GenApi::FileProtocolAdapter, [644](#)
- ReadPort
  - Spinnaker::CameraBase, [520](#)
  - Spinnaker::ICameraBase, [671](#)
- ReadRegister
  - Spinnaker::GenApi::CRegisterPortImpl, [609](#)
- Reference Interfaces, [298](#)
  - SetNumEnums, [298](#)
- RegionDestination
  - Spinnaker::Camera, [493](#)
- RegionDestination\_Stream0
  - CameraDefs Class, [119](#)
- RegionDestination\_Stream1
  - CameraDefs Class, [119](#)
- RegionDestination\_Stream2
  - CameraDefs Class, [119](#)
- RegionDestinationEnums
  - CameraDefs Class, [119](#)
- RegionMode
  - Spinnaker::Camera, [493](#)
- RegionMode\_Off
  - CameraDefs Class, [120](#)
- RegionMode\_On
  - CameraDefs Class, [120](#)
- RegionModeEnums
  - CameraDefs Class, [119](#)
- RegionSelector
  - Spinnaker::Camera, [493](#)
- RegionSelector\_All
  - CameraDefs Class, [120](#)
- RegionSelector\_Region0
  - CameraDefs Class, [120](#)
- RegionSelector\_Region1
  - CameraDefs Class, [120](#)
- RegionSelector\_Region2
  - CameraDefs Class, [120](#)
- RegionSelectorEnums
  - CameraDefs Class, [120](#)
- Register
  - NodeCallback Class, [282](#)
- RegisterCallback
  - INode Interface, [259](#)
  - Spinnaker::GenApi::Node, [777](#)
- RegisterEvent
  - Spinnaker::CameraBase, [520](#), [521](#)
  - Spinnaker::ICameraBase, [671](#)
  - Spinnaker::IInterface, [698](#)
  - Spinnaker::Interface, [741](#)

- RegisterInterfaceEvent
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::System, [819](#)
- RegisterLoggingEvent
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::System, [819](#)
- RegisterNode, [800](#)
  - Spinnaker::GenApi::RegisterNode, [802](#)
- RegisterNode Class, [299](#)
  - CRegisterRef, [299](#)
- RegisterPortImpl Class, [300](#)
- Release
  - Spinnaker::IImage, [690](#)
  - Spinnaker::Image, [719](#)
- ReleaseCameraDescriptionFileData
  - Spinnaker::GenApi::CNodeMapFactory, [588](#)
- ReleaseInstance
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::System, [819](#)
- RemovalEvent, [803](#)
  - Spinnaker::RemovalEvent, [804](#)
- RemovalEvent Class, [153](#)
- RemoveByIndex
  - Spinnaker::CameraList, [525](#)
  - Spinnaker::ICameraList, [674](#)
- RemoveBySerial
  - Spinnaker::CameraList, [525](#)
  - Spinnaker::ICameraList, [674](#)
- ReplaceEnvironmentVariables
  - GCUtilities Utility, [233](#)
- Replay
  - IPortRecorder Interface, [270](#)
  - Spinnaker::GenApi::CPortImpl, [604](#)
  - Spinnaker::GenApi::CPortWriteList, [606](#)
  - Spinnaker::GenApi::PortNode, [792](#)
  - Spinnaker::GenApi::PortReplay, [798](#)
- ReqId
  - GVCP\_REQUEST\_HEADER, [664](#)
  - U3V\_COMMAND\_HEADER, [845](#)
- Reserved
  - U3V\_EVENT\_DATA, [846](#)
- reserved
  - Spinnaker::BMPOption, [383](#)
  - Spinnaker::JPEGOption, [757](#)
  - Spinnaker::JPG2Option, [757](#)
  - Spinnaker::PGMOption, [789](#)
  - Spinnaker::PNGOption, [790](#)
  - Spinnaker::PPMOption, [799](#)
  - Spinnaker::TIFFOption, [824](#)
  - Spinnaker::Video::AVIOption, [379](#)
  - Spinnaker::Video::H264Option, [665](#)
  - Spinnaker::Video::MJPGOption, [770](#)
- ReservedOrEventSize
  - GVCP\_EVENT\_ITEM\_BASIC, [659](#)
  - GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [660](#)
  - GVCP\_EVENT\_ITEM, [658](#)
- ResetImage
  - Spinnaker::IImage, [691](#)
- Spinnaker::Image, [719](#), [720](#)
- ResetStatistics
  - Spinnaker::GenApi::CTestPortStruct, [614](#)
- resize
  - Spinnaker::GenICam::gcstring, [657](#)
- Restore
  - ISelectorDigit Interface, [275](#)
  - Spinnaker::GenApi::CSelectorSet, [611](#)
- ReverseX
  - Spinnaker::Camera, [493](#)
- ReverseY
  - Spinnaker::Camera, [493](#)
- RgbTransformLightSource
  - Spinnaker::Camera, [494](#)
- RgbTransformLightSource\_Cloudy6500K
  - CameraDefs Class, [120](#)
- RgbTransformLightSource\_CoolFluorescent4000K
  - CameraDefs Class, [120](#)
- RgbTransformLightSource\_Custom
  - CameraDefs Class, [120](#)
- RgbTransformLightSource\_Daylight5000K
  - CameraDefs Class, [120](#)
- RgbTransformLightSource\_General
  - CameraDefs Class, [120](#)
- RgbTransformLightSource\_Shade8000K
  - CameraDefs Class, [120](#)
- RgbTransformLightSource\_Tungsten2800K
  - CameraDefs Class, [120](#)
- RgbTransformLightSource\_WarmFluorescent3000K
  - CameraDefs Class, [120](#)
- RgbTransformLightSourceEnums
  - CameraDefs Class, [120](#)
- RO
  - Types Enums, [311](#)
- RW
  - Types Enums, [311](#)
- SATURATION
  - Spinnaker Definitions, [167](#)
- SET\_GUID
  - Spinnaker::GenApi, [371](#)
- SPINNAKER\_API\_ABSTRACT
  - Spinnaker Platform, [168](#)
- SPINNAKER\_API
  - Spinnaker Platform, [168](#)
- SPINNAKER\_ERR\_ABORT
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_ACCESS\_DENIED
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_BUFFER\_TOO\_SMALL
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_BUSY
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_CUSTOM\_ID
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_ERR\_ERROR
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_IM\_COLOR\_CONVERSION
  - Spinnaker Definitions, [163](#)

- SPINNAKER\_ERR\_IM\_CONVERT
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_IM\_COPY
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_IM\_DECOMPRESSION
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_ERR\_IM\_HISTOGRAM\_MEAN
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_ERR\_IM\_HISTOGRAM\_RANGE
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_ERR\_IM\_MALLOC
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_IM\_MIN\_MAX
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_ERR\_IM\_NOT\_SUPPORTED
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_INVALID\_ADDRESS
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_INVALID\_BUFFER
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_INVALID\_HANDLE
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_INVALID\_INDEX
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_INVALID\_ID
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_INVALID\_PARAMETER
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_INVALID\_VALUE
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_IO
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_NO\_DATA
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_NOT\_AVAILABLE
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_NOT\_IMPLEMENTED
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_NOT\_INITIALIZED
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_OUT\_OF\_MEMORY
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_PARSING\_CHUNK\_DATA
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_RESOURCE\_EXHAUSTED
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_RESOURCE\_IN\_USE
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_SUCCESS
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_ERR\_TIMEOUT
  - Spinnaker Definitions, [162](#)
- SPINNAKER\_EVENT\_ARRIVAL\_REMOVAL
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_EVENT\_DEVICE\_SPECIFIC
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_EVENT\_DEVICE
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_EVENT\_LOGGING\_EVENT
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_EVENT\_NEW\_BUFFER
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_EVENT\_UNKNOWN
  - Spinnaker Definitions, [163](#)
- SPINNAKER\_LOCAL
  - Spinnaker Platform, [168](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_CUST↔
  - OM\_ID
  - Spinnaker Definitions, [165](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_GEV
  - Spinnaker Definitions, [165](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_IIDC
  - Spinnaker Definitions, [165](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC↔
  - \_16BIT
  - Spinnaker Definitions, [165](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC↔
  - \_32BIT
  - Spinnaker Definitions, [165](#)
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_UNKN↔
  - OWN
  - Spinnaker Definitions, [165](#)
- SPINUPDATE\_API
  - SpinUpdate.h, [1096](#)
- STOKES\_S0\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- STOKES\_S0\_HEATMAP
  - Spinnaker Definitions, [166](#)
- STOKES\_S1\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- STOKES\_S1\_HEATMAP
  - Spinnaker Definitions, [166](#)
- STOKES\_S2\_GRAYSCALE
  - Spinnaker Definitions, [166](#)
- STOKES\_S2\_HEATMAP
  - Spinnaker Definitions, [166](#)
- Saturation
  - Spinnaker::Camera, [494](#)
- SaturationEnable
  - Spinnaker::Camera, [494](#)
- Save
  - Spinnaker::Image, [691](#)
  - Spinnaker::Image, [720–722](#)
- Scan3dAxisMax
  - Spinnaker::Camera, [494](#)
- Scan3dAxisMin
  - Spinnaker::Camera, [494](#)
- Scan3dCoordinateOffset
  - Spinnaker::Camera, [494](#)
- Scan3dCoordinateReferenceSelector
  - Spinnaker::Camera, [495](#)
- Scan3dCoordinateReferenceSelector\_RotationX
  - CameraDefs Class, [121](#)
- Scan3dCoordinateReferenceSelector\_RotationY
  - CameraDefs Class, [121](#)
- Scan3dCoordinateReferenceSelector\_RotationZ

- CameraDefs Class, [121](#)
- Scan3dCoordinateReferenceSelector\_TranslationX
  - CameraDefs Class, [121](#)
- Scan3dCoordinateReferenceSelector\_TranslationY
  - CameraDefs Class, [121](#)
- Scan3dCoordinateReferenceSelector\_TranslationZ
  - CameraDefs Class, [121](#)
- Scan3dCoordinateReferenceSelectorEnums
  - CameraDefs Class, [120](#)
- Scan3dCoordinateReferenceValue
  - Spinnaker::Camera, [495](#)
- Scan3dCoordinateScale
  - Spinnaker::Camera, [495](#)
- Scan3dCoordinateSelector
  - Spinnaker::Camera, [495](#)
- Scan3dCoordinateSelector\_CoordinateA
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSelector\_CoordinateB
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSelector\_CoordinateC
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSelectorEnums
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSystem
  - Spinnaker::Camera, [495](#)
- Scan3dCoordinateSystem\_Cartesian
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSystem\_Cylindrical
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSystem\_Spherical
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSystemEnums
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSystemReference
  - Spinnaker::Camera, [495](#)
- Scan3dCoordinateSystemReference\_Anchor
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSystemReference\_Transformed
  - CameraDefs Class, [121](#)
- Scan3dCoordinateSystemReferenceEnums
  - CameraDefs Class, [121](#)
- Scan3dCoordinateTransformSelector
  - Spinnaker::Camera, [495](#)
- Scan3dCoordinateTransformSelector\_RotationX
  - CameraDefs Class, [122](#)
- Scan3dCoordinateTransformSelector\_RotationY
  - CameraDefs Class, [122](#)
- Scan3dCoordinateTransformSelector\_RotationZ
  - CameraDefs Class, [122](#)
- Scan3dCoordinateTransformSelector\_TranslationX
  - CameraDefs Class, [122](#)
- Scan3dCoordinateTransformSelector\_TranslationY
  - CameraDefs Class, [122](#)
- Scan3dCoordinateTransformSelector\_TranslationZ
  - CameraDefs Class, [122](#)
- Scan3dCoordinateTransformSelectorEnums
  - CameraDefs Class, [121](#)
- Scan3dDistanceUnit
  - Spinnaker::Camera, [496](#)
- Scan3dDistanceUnit\_Inch
  - CameraDefs Class, [122](#)
- Scan3dDistanceUnit\_Millimeter
  - CameraDefs Class, [122](#)
- Scan3dDistanceUnitEnums
  - CameraDefs Class, [122](#)
- Scan3dInvalidDataFlag
  - Spinnaker::Camera, [496](#)
- Scan3dInvalidDataValue
  - Spinnaker::Camera, [496](#)
- Scan3dOutputMode
  - Spinnaker::Camera, [496](#)
- Scan3dOutputMode\_CalibratedABC\_Grid
  - CameraDefs Class, [122](#)
- Scan3dOutputMode\_CalibratedABC\_PointCloud
  - CameraDefs Class, [122](#)
- Scan3dOutputMode\_CalibratedAC\_Linescan
  - CameraDefs Class, [122](#)
- Scan3dOutputMode\_CalibratedAC
  - CameraDefs Class, [122](#)
- Scan3dOutputMode\_CalibratedC\_Linescan
  - CameraDefs Class, [122](#)
- Scan3dOutputMode\_CalibratedC
  - CameraDefs Class, [122](#)
- Scan3dOutputMode\_DisparityC\_Linescan
  - CameraDefs Class, [123](#)
- Scan3dOutputMode\_DisparityC
  - CameraDefs Class, [123](#)
- Scan3dOutputMode\_RectifiedC\_Linescan
  - CameraDefs Class, [122](#)
- Scan3dOutputMode\_RectifiedC
  - CameraDefs Class, [122](#)
- Scan3dOutputMode\_UncalibratedC
  - CameraDefs Class, [122](#)
- Scan3dOutputModeEnums
  - CameraDefs Class, [122](#)
- Scan3dTransformValue
  - Spinnaker::Camera, [496](#)
- SelectorSet Class, [301](#)
- SendActionCommand
  - Spinnaker::Interface, [698](#)
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::Interface, [742](#)
  - Spinnaker::System, [819](#)
- SensorDescription
  - Spinnaker::Camera, [496](#)
- SensorDigitizationTaps
  - Spinnaker::Camera, [496](#)
- SensorDigitizationTaps\_Eight
  - CameraDefs Class, [123](#)
- SensorDigitizationTaps\_Four
  - CameraDefs Class, [123](#)
- SensorDigitizationTaps\_One
  - CameraDefs Class, [123](#)
- SensorDigitizationTaps\_Ten
  - CameraDefs Class, [123](#)
- SensorDigitizationTaps\_Three



- CameraDefs Class, [123](#)
- SensorDigitizationTaps\_Two
  - CameraDefs Class, [123](#)
- SensorDigitizationTapsEnums
  - CameraDefs Class, [123](#)
- SensorHeight
  - Spinnaker::Camera, [497](#)
- SensorShutterMode
  - Spinnaker::Camera, [497](#)
- SensorShutterMode\_Global
  - CameraDefs Class, [123](#)
- SensorShutterMode\_GlobalReset
  - CameraDefs Class, [123](#)
- SensorShutterMode\_Rolling
  - CameraDefs Class, [123](#)
- SensorShutterModeEnums
  - CameraDefs Class, [123](#)
- SensorTaps
  - Spinnaker::Camera, [497](#)
- SensorTaps\_Eight
  - CameraDefs Class, [123](#)
- SensorTaps\_Four
  - CameraDefs Class, [123](#)
- SensorTaps\_One
  - CameraDefs Class, [123](#)
- SensorTaps\_Ten
  - CameraDefs Class, [123](#)
- SensorTaps\_Three
  - CameraDefs Class, [123](#)
- SensorTaps\_Two
  - CameraDefs Class, [123](#)
- SensorTapsEnums
  - CameraDefs Class, [123](#)
- SensorWidth
  - Spinnaker::Camera, [497](#)
- SequencerConfigurationMode
  - Spinnaker::Camera, [497](#)
- SequencerConfigurationMode\_Off
  - CameraDefs Class, [124](#)
- SequencerConfigurationMode\_On
  - CameraDefs Class, [124](#)
- SequencerConfigurationModeEnums
  - CameraDefs Class, [123](#)
- SequencerConfigurationValid
  - Spinnaker::Camera, [497](#)
- SequencerConfigurationValid\_No
  - CameraDefs Class, [124](#)
- SequencerConfigurationValid\_Yes
  - CameraDefs Class, [124](#)
- SequencerConfigurationValidEnums
  - CameraDefs Class, [124](#)
- SequencerFeatureEnable
  - Spinnaker::Camera, [497](#)
- SequencerMode
  - Spinnaker::Camera, [498](#)
- SequencerMode\_Off
  - CameraDefs Class, [124](#)
- SequencerMode\_On
  - CameraDefs Class, [124](#)
- SequencerModeEnums
  - CameraDefs Class, [124](#)
- SequencerPathSelector
  - Spinnaker::Camera, [498](#)
- SequencerSetActive
  - Spinnaker::Camera, [498](#)
- SequencerSetLoad
  - Spinnaker::Camera, [498](#)
- SequencerSetNext
  - Spinnaker::Camera, [498](#)
- SequencerSetSave
  - Spinnaker::Camera, [498](#)
- SequencerSetSelector
  - Spinnaker::Camera, [499](#)
- SequencerSetStart
  - Spinnaker::Camera, [499](#)
- SequencerSetValid
  - Spinnaker::Camera, [499](#)
- SequencerSetValid\_No
  - CameraDefs Class, [124](#)
- SequencerSetValid\_Yes
  - CameraDefs Class, [124](#)
- SequencerSetValidEnums
  - CameraDefs Class, [124](#)
- SequencerTriggerActivation
  - Spinnaker::Camera, [499](#)
- SequencerTriggerActivation\_AnyEdge
  - CameraDefs Class, [124](#)
- SequencerTriggerActivation\_FallingEdge
  - CameraDefs Class, [124](#)
- SequencerTriggerActivation\_LevelHigh
  - CameraDefs Class, [124](#)
- SequencerTriggerActivation\_LevelLow
  - CameraDefs Class, [124](#)
- SequencerTriggerActivation\_RisingEdge
  - CameraDefs Class, [124](#)
- SequencerTriggerActivationEnums
  - CameraDefs Class, [124](#)
- SequencerTriggerSource
  - Spinnaker::Camera, [499](#)
- SequencerTriggerSource\_FrameStart
  - CameraDefs Class, [125](#)
- SequencerTriggerSource\_Off
  - CameraDefs Class, [125](#)
- SequencerTriggerSourceEnums
  - CameraDefs Class, [124](#)
- SerialPortBaudRate
  - Spinnaker::Camera, [499](#)
- SerialPortBaudRate\_Baud115200
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud1200
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud14400
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud19200
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud230400

- CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud2400
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud300
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud38400
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud460800
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud4800
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud57600
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud600
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud921600
  - CameraDefs Class, [125](#)
- SerialPortBaudRate\_Baud9600
  - CameraDefs Class, [125](#)
- SerialPortBaudRateEnums
  - CameraDefs Class, [125](#)
- SerialPortDataBits
  - Spinnaker::Camera, [500](#)
- SerialPortParity
  - Spinnaker::Camera, [500](#)
- SerialPortParity\_Even
  - CameraDefs Class, [125](#)
- SerialPortParity\_Mark
  - CameraDefs Class, [125](#)
- SerialPortParity\_None
  - CameraDefs Class, [125](#)
- SerialPortParity\_Odd
  - CameraDefs Class, [125](#)
- SerialPortParity\_Space
  - CameraDefs Class, [125](#)
- SerialPortParityEnums
  - CameraDefs Class, [125](#)
- SerialPortSelector
  - Spinnaker::Camera, [500](#)
- SerialPortSelector\_SerialPort0
  - CameraDefs Class, [126](#)
- SerialPortSelectorEnums
  - CameraDefs Class, [125](#)
- SerialPortSource
  - Spinnaker::Camera, [500](#)
- SerialPortSource\_Line0
  - CameraDefs Class, [126](#)
- SerialPortSource\_Line1
  - CameraDefs Class, [126](#)
- SerialPortSource\_Line2
  - CameraDefs Class, [126](#)
- SerialPortSource\_Line3
  - CameraDefs Class, [126](#)
- SerialPortSource\_Off
  - CameraDefs Class, [126](#)
- SerialPortSourceEnums
  - CameraDefs Class, [126](#)
- SerialPortStopBits
  - Spinnaker::Camera, [500](#)
- SerialPortStopBits\_Bits1
  - CameraDefs Class, [126](#)
- SerialPortStopBits\_Bits1AndAHalf
  - CameraDefs Class, [126](#)
- SerialPortStopBits\_Bits2
  - CameraDefs Class, [126](#)
- SerialPortStopBitsEnums
  - CameraDefs Class, [126](#)
- SerialReceiveFramingErrorCount
  - Spinnaker::Camera, [500](#)
- SerialReceiveParityErrorCount
  - Spinnaker::Camera, [500](#)
- SerialReceiveQueueClear
  - Spinnaker::Camera, [500](#)
- SerialReceiveQueueCurrentCharacterCount
  - Spinnaker::Camera, [501](#)
- SerialReceiveQueueMaxCharacterCount
  - Spinnaker::Camera, [501](#)
- SerialTransmitQueueCurrentCharacterCount
  - Spinnaker::Camera, [501](#)
- SerialTransmitQueueMaxCharacterCount
  - Spinnaker::Camera, [501](#)
- Set
  - Spinnaker::GenApi::RegisterNode, [802](#)
- SetChannelStatus
  - Spinnaker::ImageStatistics, [696](#)
  - Spinnaker::ImageStatistics, [733](#)
- SetChunks
  - Spinnaker::ChunkData, [574](#)
  - Spinnaker::IChunkData, [680](#)
- SetCookie
  - IPortRecorder Interface, [270](#)
  - Spinnaker::GenApi::CPortWriteList, [606](#)
- SetDefaultColorProcessing
  - Spinnaker::Image, [722](#)
- SetEnumReference
  - Spinnaker::GenApi::CEnumerationTRef, [545](#)
- SetEventPayload
  - Spinnaker::Event, [637](#)
- SetEventType
  - Spinnaker::Event, [637](#)
- SetFirst
  - Spinnaker::GenApi::CSelectorSet, [611](#)
- SetGenICamCLProtocolFolder
  - GCUtilities Utility, [234](#)
- SetGenICamCacheFolder
  - GCUtilities Utility, [234](#)
- SetGenICamLogConfig
  - GCUtilities Utility, [234](#)
- SetHeatMapColorGradient
  - Spinnaker::Image, [722](#)
- SetHeatMapRange
  - Spinnaker::Image, [723](#)
- SetInfo
  - Spinnaker::GenApi::CFeatureBag, [559](#)
- SetIntValue
  - IEnumeration Interface, [246](#)



- Spinnaker::GenApi::EnumNode, [631](#)
- SetLoggingEventPriorityLevel
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::System, [820](#)
- SetMaximumFileSize
  - Spinnaker::Video::SpinVideo, [809](#)
- SetMessageCallback
  - SpinUpdate.h, [1096](#)
- SetNext
  - ISelectorDigit Interface, [276](#)
  - Spinnaker::GenApi::CSelectorSet, [611](#)
- SetNodeHandle
  - Spinnaker::GenApi::Node, [777](#)
- SetNodeMap
  - Spinnaker::GenApi::Node, [777](#)
- SetNumEnums
  - Reference Interfaces, [298](#)
  - Spinnaker::GenApi::CEnumerationTRef, [545](#)
- SetPortImpl
  - Spinnaker::GenApi::CChunkPort, [541](#)
  - Spinnaker::GenApi::CEventPort, [557](#)
  - Spinnaker::GenApi::CPortImpl, [604](#)
  - Spinnaker::GenApi::CRegisterPortImpl, [609](#)
  - Spinnaker::GenApi::PortNode, [793](#)
- SetProgressCallback
  - SpinUpdate.h, [1096](#)
- SetReference
  - Spinnaker::GenApi::BooleanNode, [385](#)
  - Spinnaker::GenApi::CEnumerationTRef, [545](#)
  - Spinnaker::GenApi::CategoryNode, [528](#)
  - Spinnaker::GenApi::CommandNode, [597](#)
  - Spinnaker::GenApi::EnumEntryNode, [627](#)
  - Spinnaker::GenApi::EnumNode, [631](#)
  - Spinnaker::GenApi::FloatNode, [649](#)
  - Spinnaker::GenApi::FloatRegNode, [652](#)
  - Spinnaker::GenApi::IntRegNode, [752](#)
  - Spinnaker::GenApi::IntegerNode, [739](#)
  - Spinnaker::GenApi::Node, [777](#)
  - Spinnaker::GenApi::PortNode, [793](#)
  - Spinnaker::GenApi::PortRecorder, [796](#)
  - Spinnaker::GenApi::PortReplay, [798](#)
  - Spinnaker::GenApi::RegisterNode, [802](#)
  - Spinnaker::GenApi::StringNode, [812](#)
  - Spinnaker::GenApi::StringRegNode, [815](#)
  - Spinnaker::GenApi::ValueNode, [849](#)
- SetValue
  - Spinnaker::GenApi::BooleanNode, [385](#)
  - Spinnaker::GenApi::CEnumerationTRef, [545](#)
  - Spinnaker::GenApi::FloatNode, [649](#)
  - Spinnaker::GenApi::IntegerNode, [739](#)
  - Spinnaker::GenApi::StringNode, [812](#)
- Sharpening
  - Spinnaker::Camera, [501](#)
- SharpeningAuto
  - Spinnaker::Camera, [501](#)
- SharpeningEnable
  - Spinnaker::Camera, [502](#)
- SharpeningThreshold
  - Spinnaker::Camera, [502](#)
- Signed
  - Types Enums, [313](#)
- SingleChunkData\_t, [805](#)
  - ChunkID, [805](#)
  - ChunkLength, [805](#)
  - ChunkOffset, [805](#)
- SingleChunkDataStr\_t, [805](#)
  - ChunkID, [805](#)
  - ChunkLength, [805](#)
  - ChunkOffset, [805](#)
- size
  - Spinnaker::GenApi::double\_autovector\_t, [619](#)
  - Spinnaker::GenApi::int64\_autovector\_t, [734](#)
  - Spinnaker::GenICam::gcstring, [657](#)
- SoftwareSignalPulse
  - Spinnaker::Camera, [502](#)
- SoftwareSignalSelector
  - Spinnaker::Camera, [502](#)
- SoftwareSignalSelector\_SoftwareSignal0
  - CameraDefs Class, [126](#)
- SoftwareSignalSelector\_SoftwareSignal1
  - CameraDefs Class, [126](#)
- SoftwareSignalSelector\_SoftwareSignal2
  - CameraDefs Class, [126](#)
- SoftwareSignalSelectorEnums
  - CameraDefs Class, [126](#)
- SourceCount
  - Spinnaker::Camera, [502](#)
- SourceSelector
  - Spinnaker::Camera, [503](#)
- SourceSelector\_All
  - CameraDefs Class, [127](#)
- SourceSelector\_Source0
  - CameraDefs Class, [127](#)
- SourceSelector\_Source1
  - CameraDefs Class, [127](#)
- SourceSelector\_Source2
  - CameraDefs Class, [127](#)
- SourceSelectorEnums
  - CameraDefs Class, [126](#)
- SpinTestCamera, [806](#)
- SpinTestCamera Class, [302](#)
- SpinUpdate.h
  - GetErrorMessage, [1096](#)
  - SPINUPDATE\_API, [1096](#)
  - SetMessageCallback, [1096](#)
  - SetProgressCallback, [1096](#)
  - UpdateFirmware, [1096](#)
  - UpdateFirmwareConsole, [1096](#)
  - UpdatorMessageCallback, [1096](#)
  - UpdatorProgressCallback, [1096](#)
- SpinVideo, [806](#)
  - Spinnaker::Video::SpinVideo, [807](#)
- Spinnaker, [319](#)
- Spinnaker Classes, [30](#)
- Spinnaker Definitions, [157](#)

- ACTION\_COMMAND\_STATUS\_ACTION\_LATE, 161
- ACTION\_COMMAND\_STATUS\_ERROR, 161
- ACTION\_COMMAND\_STATUS\_NO\_REF\_TIME, 161
- ACTION\_COMMAND\_STATUS\_OVERFLOW, 161
- ACTION\_COMMAND\_STATUS\_OK, 161
- AOP\_GRAYSCALE, 166
- AOP\_HEATMAP, 166
- ActionCommandStatus, 161
- BLUE, 167
- BMP, 163
- ColorProcessingAlgorithm, 161
- DEFAULT, 161
- DIRECTIONAL\_FILTER, 161
- DOLP\_GRAYSCALE, 166
- DOLP\_HEATMAP, 166
- EDGE\_SENSING, 161
- Error, 161
- EventType, 163
- FROM\_FILE\_EXT, 163
- FULL\_RESOLUTION, 166
- GENICAM\_ERR\_ACCESS, 162
- GENICAM\_ERR\_BAD\_ALLOCATION, 162
- GENICAM\_ERR\_DYNAMIC\_CAST, 162
- GENICAM\_ERR\_GENERIC, 162
- GENICAM\_ERR\_INVALID\_ARGUMENT, 162
- GENICAM\_ERR\_LOGICAL, 162
- GENICAM\_ERR\_OUT\_OF\_RANGE, 162
- GENICAM\_ERR\_PROPERTY, 162
- GENICAM\_ERR\_RUN\_TIME, 162
- GENICAM\_ERR\_TIMEOUT, 162
- GREEN, 167
- GREY, 167
- HEATMAP\_BLACK, 163
- HEATMAP\_BLUE, 163
- HEATMAP\_CYAN, 163
- HEATMAP\_GREEN, 163
- HEATMAP\_RED, 163
- HEATMAP\_WHITE, 163
- HEATMAP\_YELLOW, 163
- HQ\_LINEAR, 161
- HUE, 167
- HeatMapColor, 163
- IMAGE\_CHUNK\_DATA\_INVALID, 164
- IMAGE\_CRC\_CHECK\_FAILED, 164
- IMAGE\_DATA\_INCOMPLETE, 164
- IMAGE\_DATA\_OVERFLOW, 164
- IMAGE\_FILE\_FORMAT\_FORCE\_32BITS, 163
- IMAGE\_INFO\_INCONSISTENT, 164
- IMAGE\_LEADER\_BUFFER\_SIZE\_INCONSISTENT, 164
- IMAGE\_MISSING\_LEADER, 164
- IMAGE\_MISSING\_PACKETS, 164
- IMAGE\_MISSING\_TRAILER, 164
- IMAGE\_NO\_ERROR, 164
- IMAGE\_NO\_SYSTEM\_RESOURCES, 164
- IMAGE\_PACKETID\_INCONSISTENT, 164
- IMAGE\_TRAILER\_BUFFER\_SIZE\_INCONSISTENT, 164
- IMAGE\_UNKNOWN\_ERROR, 164
- IPP, 161
- ImageFileFormat, 163
- ImageStatus, 163
- IntType\_FLOAT32, 165
- IntType\_INT8, 165
- IntType\_UINT10, 165
- IntType\_UINT10P, 165
- IntType\_UINT10p, 165
- IntType\_UINT12, 165
- IntType\_UINT12P, 165
- IntType\_UINT12p, 165
- IntType\_UINT14, 165
- IntType\_UINT16, 165
- IntType\_UINT8, 165
- IntType\_UNKNOWN, 165
- JPEG12\_C, 163
- JPEG2000, 163
- JPEG, 163
- LIGHTNESS, 167
- LOG\_LEVEL\_ALERT, 166
- LOG\_LEVEL\_CRIT, 166
- LOG\_LEVEL\_DEBUG, 166
- LOG\_LEVEL\_ERROR, 166
- LOG\_LEVEL\_FATAL, 166
- LOG\_LEVEL\_INFO, 166
- LOG\_LEVEL\_NOTICE, 166
- LOG\_LEVEL\_NOTSET, 166
- LOG\_LEVEL\_OFF, 166
- LOG\_LEVEL\_WARN, 166
- NEAREST\_NEIGHBOR, 161
- NO\_COLOR\_PROCESSING, 161
- NO\_POLARIZATION, 166
- NUM\_STATISTICS\_CHANNELS, 167
- PAYLOAD\_TYPE\_CHUNK\_DATA, 164
- PAYLOAD\_TYPE\_CHUNK\_ONLY, 164
- PAYLOAD\_TYPE\_CUSTOM\_ID, 164
- PAYLOAD\_TYPE\_DEVICE\_SPECIFIC, 164
- PAYLOAD\_TYPE\_EXTENDED\_CHUNK, 164
- PAYLOAD\_TYPE\_FILE, 164
- PAYLOAD\_TYPE\_H264, 164
- PAYLOAD\_TYPE\_IMAGE, 164
- PAYLOAD\_TYPE\_JPEG2000, 164
- PAYLOAD\_TYPE\_JPEG, 164
- PAYLOAD\_TYPE\_MULTI\_PART, 164
- PAYLOAD\_TYPE\_RAW\_DATA, 164
- PAYLOAD\_TYPE\_UNKNOWN, 164
- PGM, 163
- PNG, 163
- PPM, 163
- PayloadTypeInfoIDs, 164
- PixelFormatIntType, 164
- PixelFormatNamespaceID, 165
- PolarizationAlgorithm, 165
- PolarizationResolution, 166

- QUADRANT\_I0\_GRAYSCALE, 166
- QUADRANT\_I135\_GRAYSCALE, 166
- QUADRANT\_I45\_GRAYSCALE, 166
- QUADRANT\_I90\_GRAYSCALE, 166
- QUARTER\_RESOLUTION, 166
- RAW, 163
- RED, 167
- RIGOROUS, 161
- SATURATION, 167
- SPINNAKER\_ERR\_ABORT, 162
- SPINNAKER\_ERR\_ACCESS\_DENIED, 162
- SPINNAKER\_ERR\_BUFFER\_TOO\_SMALL, 162
- SPINNAKER\_ERR\_BUSY, 162
- SPINNAKER\_ERR\_CUSTOM\_ID, 163
- SPINNAKER\_ERR\_ERROR, 162
- SPINNAKER\_ERR\_IM\_COLOR\_CONVERSION, 163
- SPINNAKER\_ERR\_IM\_CONVERT, 162
- SPINNAKER\_ERR\_IM\_COPY, 162
- SPINNAKER\_ERR\_IM\_DECOMPRESSION, 163
- SPINNAKER\_ERR\_IM\_HISTOGRAM\_MEAN, 163
- SPINNAKER\_ERR\_IM\_HISTOGRAM\_RANGE, 163
- SPINNAKER\_ERR\_IM\_MALLOC, 162
- SPINNAKER\_ERR\_IM\_MIN\_MAX, 163
- SPINNAKER\_ERR\_IM\_NOT\_SUPPORTED, 162
- SPINNAKER\_ERR\_INVALID\_ADDRESS, 162
- SPINNAKER\_ERR\_INVALID\_BUFFER, 162
- SPINNAKER\_ERR\_INVALID\_HANDLE, 162
- SPINNAKER\_ERR\_INVALID\_INDEX, 162
- SPINNAKER\_ERR\_INVALID\_ID, 162
- SPINNAKER\_ERR\_INVALID\_PARAMETER, 162
- SPINNAKER\_ERR\_INVALID\_VALUE, 162
- SPINNAKER\_ERR\_IO, 162
- SPINNAKER\_ERR\_NO\_DATA, 162
- SPINNAKER\_ERR\_NOT\_AVAILABLE, 162
- SPINNAKER\_ERR\_NOT\_IMPLEMENTED, 162
- SPINNAKER\_ERR\_NOT\_INITIALIZED, 162
- SPINNAKER\_ERR\_OUT\_OF\_MEMORY, 162
- SPINNAKER\_ERR\_PARSING\_CHUNK\_DATA, 162
- SPINNAKER\_ERR\_RESOURCE\_EXHAUSTED, 162
- SPINNAKER\_ERR\_RESOURCE\_IN\_USE, 162
- SPINNAKER\_ERR\_SUCCESS, 162
- SPINNAKER\_ERR\_TIMEOUT, 162
- SPINNAKER\_EVENT\_ARRIVAL\_REMOVAL, 163
- SPINNAKER\_EVENT\_DEVICE\_SPECIFIC, 163
- SPINNAKER\_EVENT\_DEVICE, 163
- SPINNAKER\_EVENT\_LOGGING\_EVENT, 163
- SPINNAKER\_EVENT\_NEW\_BUFFER, 163
- SPINNAKER\_EVENT\_UNKNOWN, 163
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_CUSTOM\_ID, 165
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_GEV, 165
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_IIDC, 165
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PNC\_16BIT, 165
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PNC\_32BIT, 165
- SPINNAKER\_PIXELFORMAT\_NAMESPACE\_UNKNOWN, 165
- STOKES\_S0\_GRAYSCALE, 166
- STOKES\_S0\_HEATMAP, 166
- STOKES\_S1\_GRAYSCALE, 166
- STOKES\_S1\_HEATMAP, 166
- STOKES\_S2\_GRAYSCALE, 166
- STOKES\_S2\_HEATMAP, 166
- SpinnakerLogLevel, 166
- StatisticsChannel, 166
- TIFF, 163
- WEIGHTED\_DIRECTIONAL\_FILTER, 161
- Spinnaker Event Classes, 27
- Spinnaker GenApi Classes, 190
  - \_ClearXMLCache, 196
  - \_Connect, 196
  - \_Destroy, 196
  - \_GetDeviceName, 196
  - \_GetNode, 196
  - \_GetNodes, 196
  - \_GetSupportedSchemaVersions, 196
  - \_InvalidateNodes, 196
  - \_LoadXMLFromFile, 196
  - \_LoadXMLFromFileInject, 196
  - \_LoadXMLFromString, 197
  - \_LoadXMLFromStringInject, 197
  - \_LoadXMLFromZIPData, 197
  - \_LoadXMLFromZIPFile, 197
  - \_Poll, 197
  - ~CNodeMapRefT, 198
  - CNodeMapRef, 196
  - CNodeMapRefT, 197
  - CNodeRef, 196
  - CSelectorRef, 196
  - CastToIDestroy, 197
  - EatComments, 197
  - operator<<, 197
  - operator>>, 198
  - operator=, 197
- Spinnaker GenApi Enums, 307
- Spinnaker GenApi Interfaces, 200
  - CallbackHandleType, 201
  - NodeList\_t, 201
- Spinnaker GenApi Utilities, 230
- Spinnaker Headers, 154
  - EVENT\_TIMEOUT\_INFINITE, 155
  - EVENT\_TIMEOUT\_NONE, 155
- Spinnaker Platform, 168
  - SPINNAKER\_API\_ABSTRACT, 168
  - SPINNAKER\_API, 168
  - SPINNAKER\_LOCAL, 168
- Spinnaker QuickSpin Classes, 173
- Spinnaker Video Class, 169
- Spinnaker Video Definitions, 170

- Spinnaker.h, [156](#)
- Spinnaker::ActionCommandResult
  - DeviceAddress, [375](#)
  - Status, [375](#)
- Spinnaker::ArrivalEvent
  - ~ArrivalEvent, [377](#)
  - ArrivalEvent, [377](#)
  - OnDeviceArrival, [377](#)
  - operator=, [377](#)
- Spinnaker::BMPOption
  - BMPOption, [383](#)
  - indexedColor\_8bit, [383](#)
  - reserved, [383](#)
- Spinnaker::BasePtr
  - ~BasePtr, [381](#)
  - BasePtr, [381](#)
  - get, [381](#)
  - IsValid, [381](#)
  - m\_pT, [382](#)
  - operator bool, [381](#)
  - operator T \*, [381](#)
  - operator->, [381](#)
  - operator=, [381](#)
  - operator==, [381](#), [382](#)
- Spinnaker::Camera
  - ~Camera, [416](#)
  - aPAUSEMACCtrlFramesReceived, [420](#)
  - aPAUSEMACCtrlFramesTransmitted, [420](#)
  - AasRoiEnable, [416](#)
  - AasRoiHeight, [416](#)
  - AasRoiOffsetX, [416](#)
  - AasRoiOffsetY, [416](#)
  - AasRoiWidth, [417](#)
  - AcquisitionAbort, [417](#)
  - AcquisitionArm, [417](#)
  - AcquisitionBurstFrameCount, [417](#)
  - AcquisitionFrameCount, [417](#)
  - AcquisitionFrameRate, [418](#)
  - AcquisitionFrameRateEnable, [418](#)
  - AcquisitionLineRate, [418](#)
  - AcquisitionMode, [418](#)
  - AcquisitionResultingFrameRate, [418](#)
  - AcquisitionStart, [418](#)
  - AcquisitionStatus, [418](#)
  - AcquisitionStatusSelector, [419](#)
  - AcquisitionStop, [419](#)
  - ActionDeviceKey, [419](#)
  - ActionGroupKey, [419](#)
  - ActionGroupMask, [419](#)
  - ActionQueueSize, [419](#)
  - ActionSelector, [419](#)
  - ActionUnconditionalMode, [420](#)
  - AdaptiveCompressionEnable, [420](#)
  - AdcBitDepth, [420](#)
  - AutoAlgorithmSelector, [420](#)
  - AutoExposureControlLoopDamping, [420](#)
  - AutoExposureControlPriority, [421](#)
  - AutoExposureEVCompensation, [421](#)
  - AutoExposureExposureTimeLowerLimit, [421](#)
  - AutoExposureExposureTimeUpperLimit, [421](#)
  - AutoExposureGainLowerLimit, [421](#)
  - AutoExposureGainUpperLimit, [422](#)
  - AutoExposureGreyValueLowerLimit, [422](#)
  - AutoExposureGreyValueUpperLimit, [422](#)
  - AutoExposureLightingMode, [422](#)
  - AutoExposureMeteringMode, [422](#)
  - AutoExposureTargetGreyValue, [423](#)
  - AutoExposureTargetGreyValueAuto, [423](#)
  - BalanceRatio, [423](#)
  - BalanceRatioSelector, [423](#)
  - BalanceWhiteAuto, [424](#)
  - BalanceWhiteAutoDamping, [424](#)
  - BalanceWhiteAutoLowerLimit, [424](#)
  - BalanceWhiteAutoProfile, [424](#)
  - BalanceWhiteAutoUpperLimit, [424](#)
  - BinningHorizontal, [425](#)
  - BinningHorizontalMode, [425](#)
  - BinningSelector, [425](#)
  - BinningVertical, [425](#)
  - BinningVerticalMode, [425](#)
  - BlackLevel, [425](#)
  - BlackLevelAuto, [426](#)
  - BlackLevelAutoBalance, [426](#)
  - BlackLevelClampingEnable, [426](#)
  - BlackLevelRaw, [426](#)
  - BlackLevelSelector, [426](#)
  - Camera, [416](#)
  - ChunkBlackLevel, [426](#)
  - ChunkBlackLevelSelector, [427](#)
  - ChunkCRC, [427](#)
  - ChunkCounterSelector, [427](#)
  - ChunkCounterValue, [427](#)
  - ChunkEnable, [427](#)
  - ChunkEncoderSelector, [427](#)
  - ChunkEncoderStatus, [427](#)
  - ChunkEncoderValue, [427](#)
  - ChunkExposureEndLineStatusAll, [428](#)
  - ChunkExposureTime, [428](#)
  - ChunkExposureTimeSelector, [428](#)
  - ChunkFrameID, [428](#)
  - ChunkGain, [428](#)
  - ChunkGainSelector, [428](#)
  - ChunkHeight, [428](#)
  - ChunkImage, [428](#)
  - ChunkImageComponent, [429](#)
  - ChunkInferenceConfidence, [429](#)
  - ChunkInferenceResult, [429](#)
  - ChunkLinePitch, [429](#)
  - ChunkLineStatusAll, [429](#)
  - ChunkModeActive, [429](#)
  - ChunkOffsetX, [429](#)
  - ChunkOffsetY, [429](#)
  - ChunkPartSelector, [430](#)
  - ChunkPixelDynamicRangeMax, [430](#)
  - ChunkPixelDynamicRangeMin, [430](#)
  - ChunkPixelFormat, [430](#)

- ChunkRegionID, [430](#)
- ChunkScan3dAxisMax, [430](#)
- ChunkScan3dAxisMin, [430](#)
- ChunkScan3dCoordinateOffset, [430](#)
- ChunkScan3dCoordinateReferenceSelector, [431](#)
- ChunkScan3dCoordinateReferenceValue, [431](#)
- ChunkScan3dCoordinateScale, [431](#)
- ChunkScan3dCoordinateSelector, [431](#)
- ChunkScan3dCoordinateSystem, [431](#)
- ChunkScan3dCoordinateSystemReference, [431](#)
- ChunkScan3dCoordinateTransformSelector, [431](#)
- ChunkScan3dDistanceUnit, [432](#)
- ChunkScan3dInvalidDataFlag, [432](#)
- ChunkScan3dInvalidDataValue, [432](#)
- ChunkScan3dOutputMode, [432](#)
- ChunkScan3dTransformValue, [432](#)
- ChunkScanLineSelector, [432](#)
- ChunkSelector, [432](#)
- ChunkSequencerSetActive, [433](#)
- ChunkSerialData, [433](#)
- ChunkSerialDataLength, [433](#)
- ChunkSerialReceiveOverflow, [433](#)
- ChunkSourceID, [433](#)
- ChunkStreamChannelID, [433](#)
- ChunkTimerSelector, [433](#)
- ChunkTimerValue, [433](#)
- ChunkTimestamp, [434](#)
- ChunkTimestampLatchValue, [434](#)
- ChunkTransferBlockID, [434](#)
- ChunkTransferQueueCurrentBlockCount, [434](#)
- ChunkTransferStreamID, [434](#)
- ChunkWidth, [434](#)
- CIConfiguration, [434](#)
- CITimeSlotsCount, [435](#)
- ColorTransformationEnable, [435](#)
- ColorTransformationSelector, [435](#)
- ColorTransformationValue, [435](#)
- ColorTransformationValueSelector, [435](#)
- CompressionRatio, [435](#)
- CounterDelay, [436](#)
- CounterDuration, [436](#)
- CounterEventActivation, [436](#)
- CounterEventSource, [436](#)
- CounterReset, [436](#)
- CounterResetActivation, [436](#)
- CounterResetSource, [436](#)
- CounterSelector, [436](#)
- CounterStatus, [437](#)
- CounterTriggerActivation, [437](#)
- CounterTriggerSource, [437](#)
- CounterValue, [437](#)
- CounterValueAtReset, [437](#)
- CxpConnectionSelector, [437](#)
- CxpConnectionTestErrorCount, [437](#)
- CxpConnectionTestMode, [437](#)
- CxpConnectionTestPacketCount, [438](#)
- CxpLinkConfiguration, [438](#)
- CxpLinkConfigurationPreferred, [438](#)
- CxpLinkConfigurationStatus, [438](#)
- CxpPoCxpAuto, [438](#)
- CxpPoCxpStatus, [438](#)
- CxpPoCxpTripReset, [438](#)
- CxpPoCxpTurnOff, [439](#)
- DecimationHorizontal, [439](#)
- DecimationHorizontalMode, [439](#)
- DecimationSelector, [439](#)
- DecimationVertical, [439](#)
- DecimationVerticalMode, [440](#)
- DefectCorrectStaticEnable, [440](#)
- DefectCorrectionMode, [440](#)
- DefectTableApply, [440](#)
- DefectTableCoordinateX, [440](#)
- DefectTableCoordinateY, [441](#)
- DefectTableFactoryRestore, [441](#)
- DefectTableIndex, [441](#)
- DefectTablePixelCount, [441](#)
- DefectTableSave, [441](#)
- Deinterlacing, [442](#)
- DeviceCharacterSet, [442](#)
- DeviceClockFrequency, [442](#)
- DeviceClockSelector, [442](#)
- DeviceConnectionSelector, [442](#)
- DeviceConnectionSpeed, [442](#)
- DeviceConnectionStatus, [442](#)
- DeviceEventChannelCount, [443](#)
- DeviceFamilyName, [443](#)
- DeviceFeaturePersistenceEnd, [443](#)
- DeviceFeaturePersistenceStart, [443](#)
- DeviceFirmwareVersion, [443](#)
- DeviceGenCPVersionMajor, [443](#)
- DeviceGenCPVersionMinor, [443](#)
- DeviceID, [444](#)
- DeviceIndicatorMode, [444](#)
- DeviceLinkBandwidthReserve, [444](#)
- DeviceLinkCommandTimeout, [444](#)
- DeviceLinkConnectionCount, [444](#)
- DeviceLinkCurrentThroughput, [444](#)
- DeviceLinkHeartbeatMode, [444](#)
- DeviceLinkHeartbeatTimeout, [445](#)
- DeviceLinkSelector, [445](#)
- DeviceLinkSpeed, [445](#)
- DeviceLinkThroughputLimit, [445](#)
- DeviceLinkThroughputLimitMode, [445](#)
- DeviceManifestEntrySelector, [445](#)
- DeviceManifestPrimaryURL, [446](#)
- DeviceManifestSchemaMajorVersion, [446](#)
- DeviceManifestSchemaMinorVersion, [446](#)
- DeviceManifestSecondaryURL, [446](#)
- DeviceManifestXMLMajorVersion, [446](#)
- DeviceManifestXMLMinorVersion, [446](#)
- DeviceManifestXMLSubMinorVersion, [446](#)
- DeviceManufacturerInfo, [446](#)
- DeviceMaxThroughput, [447](#)
- DeviceModelName, [447](#)
- DevicePowerSupplySelector, [447](#)
- DeviceRegistersCheck, [447](#)

- DeviceRegistersEndianness, [447](#)
- DeviceRegistersStreamingEnd, [447](#)
- DeviceRegistersStreamingStart, [448](#)
- DeviceRegistersValid, [448](#)
- DeviceReset, [448](#)
- DeviceSFNCVersionMajor, [449](#)
- DeviceSFNCVersionMinor, [449](#)
- DeviceSFNCVersionSubMinor, [449](#)
- DeviceScanType, [448](#)
- DeviceSerialNumber, [448](#)
- DeviceSerialPortBaudRate, [448](#)
- DeviceSerialPortSelector, [448](#)
- DeviceStreamChannelCount, [449](#)
- DeviceStreamChannelEndianness, [449](#)
- DeviceStreamChannelLink, [449](#)
- DeviceStreamChannelPacketSize, [449](#)
- DeviceStreamChannelSelector, [450](#)
- DeviceStreamChannelType, [450](#)
- DeviceTLType, [450](#)
- DeviceTLVersionMajor, [450](#)
- DeviceTLVersionMinor, [451](#)
- DeviceTLVersionSubMinor, [451](#)
- DeviceTapGeometry, [450](#)
- DeviceTemperature, [450](#)
- DeviceTemperatureSelector, [450](#)
- DeviceType, [451](#)
- DeviceUptime, [451](#)
- DeviceUserID, [451](#)
- DeviceVendorName, [451](#)
- DeviceVersion, [451](#)
- EncoderDivider, [452](#)
- EncoderMode, [452](#)
- EncoderOutputMode, [452](#)
- EncoderReset, [452](#)
- EncoderResetActivation, [452](#)
- EncoderResetSource, [452](#)
- EncoderSelector, [452](#)
- EncoderSourceA, [453](#)
- EncoderSourceB, [453](#)
- EncoderStatus, [453](#)
- EncoderTimeout, [453](#)
- EncoderValue, [453](#)
- EncoderValueAtReset, [453](#)
- EnumerationCount, [453](#)
- EventAcquisitionEnd, [454](#)
- EventAcquisitionEndFrameID, [454](#)
- EventAcquisitionEndTimestamp, [454](#)
- EventAcquisitionError, [454](#)
- EventAcquisitionErrorFrameID, [454](#)
- EventAcquisitionErrorTimestamp, [454](#)
- EventAcquisitionStart, [454](#)
- EventAcquisitionStartFrameID, [454](#)
- EventAcquisitionStartTimestamp, [455](#)
- EventAcquisitionTransferEnd, [455](#)
- EventAcquisitionTransferEndFrameID, [455](#)
- EventAcquisitionTransferEndTimestamp, [455](#)
- EventAcquisitionTransferStart, [455](#)
- EventAcquisitionTransferStartFrameID, [455](#)
- EventAcquisitionTransferStartTimestamp, [455](#)
- EventAcquisitionTrigger, [456](#)
- EventAcquisitionTriggerFrameID, [456](#)
- EventAcquisitionTriggerTimestamp, [456](#)
- EventActionLate, [456](#)
- EventActionLateFrameID, [456](#)
- EventActionLateTimestamp, [456](#)
- EventCounter0End, [456](#)
- EventCounter0EndFrameID, [456](#)
- EventCounter0EndTimestamp, [457](#)
- EventCounter0Start, [457](#)
- EventCounter0StartFrameID, [457](#)
- EventCounter0StartTimestamp, [457](#)
- EventCounter1End, [457](#)
- EventCounter1EndFrameID, [457](#)
- EventCounter1EndTimestamp, [457](#)
- EventCounter1Start, [457](#)
- EventCounter1StartFrameID, [458](#)
- EventCounter1StartTimestamp, [458](#)
- EventEncoder0Restarted, [458](#)
- EventEncoder0RestartedFrameID, [458](#)
- EventEncoder0RestartedTimestamp, [458](#)
- EventEncoder0Stopped, [458](#)
- EventEncoder0StoppedFrameID, [458](#)
- EventEncoder0StoppedTimestamp, [458](#)
- EventEncoder1Restarted, [459](#)
- EventEncoder1RestartedFrameID, [459](#)
- EventEncoder1RestartedTimestamp, [459](#)
- EventEncoder1Stopped, [459](#)
- EventEncoder1StoppedFrameID, [459](#)
- EventEncoder1StoppedTimestamp, [459](#)
- EventError, [459](#)
- EventErrorCode, [459](#)
- EventErrorFrameID, [460](#)
- EventErrorTimestamp, [460](#)
- EventExposureEnd, [460](#)
- EventExposureEndFrameID, [460](#)
- EventExposureEndTimestamp, [460](#)
- EventExposureStart, [460](#)
- EventExposureStartFrameID, [460](#)
- EventExposureStartTimestamp, [460](#)
- EventFrameBurstEnd, [461](#)
- EventFrameBurstEndFrameID, [461](#)
- EventFrameBurstEndTimestamp, [461](#)
- EventFrameBurstStart, [461](#)
- EventFrameBurstStartFrameID, [461](#)
- EventFrameBurstStartTimestamp, [461](#)
- EventFrameEnd, [461](#)
- EventFrameEndFrameID, [461](#)
- EventFrameEndTimestamp, [462](#)
- EventFrameStart, [462](#)
- EventFrameStartFrameID, [462](#)
- EventFrameStartTimestamp, [462](#)
- EventFrameTransferEnd, [462](#)
- EventFrameTransferEndFrameID, [462](#)
- EventFrameTransferEndTimestamp, [462](#)
- EventFrameTransferStart, [462](#)
- EventFrameTransferStartFrameID, [463](#)



- EventFrameTransferStartTimestamp, [463](#)
- EventFrameTrigger, [463](#)
- EventFrameTriggerFrameID, [463](#)
- EventFrameTriggerTimestamp, [463](#)
- EventLine0AnyEdge, [463](#)
- EventLine0AnyEdgeFrameID, [463](#)
- EventLine0AnyEdgeTimestamp, [464](#)
- EventLine0FallingEdge, [464](#)
- EventLine0FallingEdgeFrameID, [464](#)
- EventLine0FallingEdgeTimestamp, [464](#)
- EventLine0RisingEdge, [464](#)
- EventLine0RisingEdgeFrameID, [464](#)
- EventLine0RisingEdgeTimestamp, [464](#)
- EventLine1AnyEdge, [464](#)
- EventLine1AnyEdgeFrameID, [465](#)
- EventLine1AnyEdgeTimestamp, [465](#)
- EventLine1FallingEdge, [465](#)
- EventLine1FallingEdgeFrameID, [465](#)
- EventLine1FallingEdgeTimestamp, [465](#)
- EventLine1RisingEdge, [465](#)
- EventLine1RisingEdgeFrameID, [465](#)
- EventLine1RisingEdgeTimestamp, [465](#)
- EventLinkSpeedChange, [466](#)
- EventLinkSpeedChangeFrameID, [466](#)
- EventLinkSpeedChangeTimestamp, [466](#)
- EventLinkTrigger0, [466](#)
- EventLinkTrigger0FrameID, [466](#)
- EventLinkTrigger0Timestamp, [466](#)
- EventLinkTrigger1, [466](#)
- EventLinkTrigger1FrameID, [466](#)
- EventLinkTrigger1Timestamp, [467](#)
- EventNotification, [467](#)
- EventSelector, [467](#)
- EventSequencerSetChange, [467](#)
- EventSequencerSetChangeFrameID, [467](#)
- EventSequencerSetChangeTimestamp, [467](#)
- EventSerialData, [467](#)
- EventSerialDataLength, [467](#)
- EventSerialPortReceive, [468](#)
- EventSerialPortReceiveTimestamp, [468](#)
- EventSerialReceiveOverflow, [468](#)
- EventStream0TransferBlockEnd, [468](#)
- EventStream0TransferBlockEndFrameID, [468](#)
- EventStream0TransferBlockEndTimestamp, [468](#)
- EventStream0TransferBlockStart, [468](#)
- EventStream0TransferBlockStartFrameID, [469](#)
- EventStream0TransferBlockStartTimestamp, [469](#)
- EventStream0TransferBlockTrigger, [469](#)
- EventStream0TransferBlockTriggerFrameID, [469](#)
- EventStream0TransferBlockTriggerTimestamp, [469](#)
- EventStream0TransferBurstEnd, [469](#)
- EventStream0TransferBurstEndFrameID, [469](#)
- EventStream0TransferBurstEndTimestamp, [470](#)
- EventStream0TransferBurstStart, [470](#)
- EventStream0TransferBurstStartFrameID, [470](#)
- EventStream0TransferBurstStartTimestamp, [470](#)
- EventStream0TransferEnd, [470](#)
- EventStream0TransferEndFrameID, [470](#)
- EventStream0TransferEndTimestamp, [470](#)
- EventStream0TransferOverflow, [471](#)
- EventStream0TransferOverflowFrameID, [471](#)
- EventStream0TransferOverflowTimestamp, [471](#)
- EventStream0TransferPause, [471](#)
- EventStream0TransferPauseFrameID, [471](#)
- EventStream0TransferPauseTimestamp, [471](#)
- EventStream0TransferResume, [471](#)
- EventStream0TransferResumeFrameID, [472](#)
- EventStream0TransferResumeTimestamp, [472](#)
- EventStream0TransferStart, [472](#)
- EventStream0TransferStartFrameID, [472](#)
- EventStream0TransferStartTimestamp, [472](#)
- EventTest, [472](#)
- EventTestTimestamp, [472](#)
- EventTimer0End, [473](#)
- EventTimer0EndFrameID, [473](#)
- EventTimer0EndTimestamp, [473](#)
- EventTimer0Start, [473](#)
- EventTimer0StartFrameID, [473](#)
- EventTimer0StartTimestamp, [473](#)
- EventTimer1End, [473](#)
- EventTimer1EndFrameID, [473](#)
- EventTimer1EndTimestamp, [474](#)
- EventTimer1Start, [474](#)
- EventTimer1StartFrameID, [474](#)
- EventTimer1StartTimestamp, [474](#)
- ExposureActiveMode, [474](#)
- ExposureAuto, [474](#)
- ExposureMode, [474](#)
- ExposureTime, [474](#)
- ExposureTimeMode, [475](#)
- ExposureTimeSelector, [475](#)
- FactoryReset, [475](#)
- FileAccessBuffer, [475](#)
- FileAccessLength, [475](#)
- FileAccessOffset, [475](#)
- FileOpenMode, [475](#)
- FileOperationExecute, [476](#)
- FileOperationResult, [476](#)
- FileOperationSelector, [476](#)
- FileOperationStatus, [476](#)
- FileSelector, [476](#)
- FileSize, [476](#)
- Gain, [477](#)
- GainAuto, [477](#)
- GainAutoBalance, [477](#)
- GainSelector, [477](#)
- Gamma, [477](#)
- GammaEnable, [477](#)
- GevActiveLinkCount, [477](#)
- GevCCP, [478](#)
- GevCurrentDefaultGateway, [478](#)
- GevCurrentIPAddress, [478](#)
- GevCurrentIPConfigurationDHCP, [478](#)
- GevCurrentIPConfigurationLLA, [478](#)
- GevCurrentIPConfigurationPersistentIP, [478](#)
- GevCurrentPhysicalLinkConfiguration, [478](#)

- GevCurrentSubnetMask, [478](#)
- GevDiscoveryAckDelay, [479](#)
- GevFirstURL, [479](#)
- GevGVCPExtendedStatusCodes, [479](#)
- GevGVCPExtendedStatusCodesSelector, [479](#)
- GevGVCPHeartbeatDisable, [479](#)
- GevGVCPPendingAck, [479](#)
- GevGVCPPendingTimeout, [479](#)
- GevGVSPExtendedIDMode, [480](#)
- GevHeartbeatTimeout, [480](#)
- GevIEEE1588, [480](#)
- GevIEEE1588ClockAccuracy, [480](#)
- GevIEEE1588Mode, [480](#)
- GevIEEE1588Status, [480](#)
- GevIPConfigurationStatus, [481](#)
- GevInterfaceSelector, [480](#)
- GevMACAddress, [481](#)
- GevMCDA, [481](#)
- GevMCPHostPort, [481](#)
- GevMCRC, [481](#)
- GevMCSP, [481](#)
- GevMCTT, [481](#)
- GevNumberOfInterfaces, [481](#)
- GevPAUSEFrameReception, [482](#)
- GevPAUSEFrameTransmission, [482](#)
- GevPersistentDefaultGateway, [482](#)
- GevPersistentIPAddress, [482](#)
- GevPersistentSubnetMask, [482](#)
- GevPhysicalLinkConfiguration, [482](#)
- GevPrimaryApplicationIPAddress, [482](#)
- GevPrimaryApplicationSocket, [482](#)
- GevPrimaryApplicationSwitchoverKey, [483](#)
- GevSCCFGAllInTransmission, [483](#)
- GevSCCFGExtendedChunkData, [483](#)
- GevSCCFGPacketResendDestination, [483](#)
- GevSCCFGUnconditionalStreaming, [483](#)
- GevSCDA, [483](#)
- GevSCPDDirection, [484](#)
- GevSCPHostPort, [484](#)
- GevSCPInterfaceIndex, [484](#)
- GevSCPSBigEndian, [484](#)
- GevSCPSDoNotFragment, [484](#)
- GevSCPSFireTestPacket, [484](#)
- GevSCPSPacketSize, [484](#)
- GevSCPD, [483](#)
- GevSCSP, [485](#)
- GevSCZoneConfigurationLock, [485](#)
- GevSCZoneCount, [485](#)
- GevSCZoneDirectionAll, [485](#)
- GevSecondURL, [485](#)
- GevStreamChannelSelector, [485](#)
- GevSupportedOption, [485](#)
- GevSupportedOptionSelector, [486](#)
- GevTimestampTickFrequency, [486](#)
- GuiXmlManifestAddress, [486](#)
- Height, [486](#)
- HeightMax, [486](#)
- ImageComponentEnable, [486](#)
- ImageComponentSelector, [486](#)
- ImageCompressionBitrate, [487](#)
- ImageCompressionJPEGFormatOption, [487](#)
- ImageCompressionMode, [487](#)
- ImageCompressionQuality, [487](#)
- ImageCompressionRateOption, [487](#)
- Init, [416](#)
- IspEnable, [487](#)
- LUTEnable, [490](#)
- LUTIndex, [490](#)
- LUTSelector, [490](#)
- LUTValue, [490](#)
- LUTValueAll, [491](#)
- LineFilterWidth, [487](#)
- LineFormat, [488](#)
- LineInputFilterSelector, [488](#)
- LineInverter, [488](#)
- LineMode, [488](#)
- LinePitch, [488](#)
- LineSelector, [488](#)
- LineSource, [488](#)
- LineStatus, [488](#)
- LineStatusAll, [489](#)
- LinkErrorCount, [489](#)
- LinkUptime, [489](#)
- LogicBlockLUTInputActivation, [489](#)
- LogicBlockLUTInputSelector, [489](#)
- LogicBlockLUTInputSource, [489](#)
- LogicBlockLUTOutputValue, [489](#)
- LogicBlockLUTOutputValueAll, [489](#)
- LogicBlockLUTRowIndex, [490](#)
- LogicBlockLUTSelector, [490](#)
- LogicBlockSelector, [490](#)
- MaxDeviceResetTime, [491](#)
- OffsetX, [491](#)
- OffsetY, [491](#)
- PacketResendRequestCount, [491](#)
- PayloadSize, [491](#)
- PixelColorFilter, [491](#)
- PixelDynamicRangeMax, [492](#)
- PixelDynamicRangeMin, [492](#)
- PixelFormat, [492](#)
- PixelFormatInfoID, [492](#)
- PixelFormatInfoSelector, [492](#)
- PixelSize, [492](#)
- PowerSupplyCurrent, [492](#)
- PowerSupplyVoltage, [493](#)
- RegionDestination, [493](#)
- RegionMode, [493](#)
- RegionSelector, [493](#)
- ReverseX, [493](#)
- ReverseY, [493](#)
- RgbTransformLightSource, [494](#)
- Saturation, [494](#)
- SaturationEnable, [494](#)
- Scan3dAxisMax, [494](#)
- Scan3dAxisMin, [494](#)
- Scan3dCoordinateOffset, [494](#)



Scan3dCoordinateReferenceSelector, 495  
Scan3dCoordinateReferenceValue, 495  
Scan3dCoordinateScale, 495  
Scan3dCoordinateSelector, 495  
Scan3dCoordinateSystem, 495  
Scan3dCoordinateSystemReference, 495  
Scan3dCoordinateTransformSelector, 495  
Scan3dDistanceUnit, 496  
Scan3dInvalidDataFlag, 496  
Scan3dInvalidDataValue, 496  
Scan3dOutputMode, 496  
Scan3dTransformValue, 496  
SensorDescription, 496  
SensorDigitizationTaps, 496  
SensorHeight, 497  
SensorShutterMode, 497  
SensorTaps, 497  
SensorWidth, 497  
SequencerConfigurationMode, 497  
SequencerConfigurationValid, 497  
SequencerFeatureEnable, 497  
SequencerMode, 498  
SequencerPathSelector, 498  
SequencerSetActive, 498  
SequencerSetLoad, 498  
SequencerSetNext, 498  
SequencerSetSave, 498  
SequencerSetSelector, 499  
SequencerSetStart, 499  
SequencerSetValid, 499  
SequencerTriggerActivation, 499  
SequencerTriggerSource, 499  
SerialPortBaudRate, 499  
SerialPortDataBits, 500  
SerialPortParity, 500  
SerialPortSelector, 500  
SerialPortSource, 500  
SerialPortStopBits, 500  
SerialReceiveFramingErrorCount, 500  
SerialReceiveParityErrorCount, 500  
SerialReceiveQueueClear, 500  
SerialReceiveQueueCurrentCharacterCount, 501  
SerialReceiveQueueMaxCharacterCount, 501  
SerialTransmitQueueCurrentCharacterCount, 501  
SerialTransmitQueueMaxCharacterCount, 501  
Sharpening, 501  
SharpeningAuto, 501  
SharpeningEnable, 502  
SharpeningThreshold, 502  
SoftwareSignalPulse, 502  
SoftwareSignalSelector, 502  
SourceCount, 502  
SourceSelector, 503  
TLParamsLocked, 505  
Test0001, 503  
TestEventGenerate, 503  
TestPattern, 503  
TestPatternGeneratorSelector, 503  
TestPendingAck, 503  
TimerDelay, 503  
TimerDuration, 504  
TimerReset, 504  
TimerSelector, 504  
TimerStatus, 504  
TimerTriggerActivation, 504  
TimerTriggerSource, 504  
TimerValue, 504  
Timestamp, 505  
TimestampLatch, 505  
TimestampLatchValue, 505  
TimestampReset, 505  
TransferAbort, 505  
TransferBlockCount, 505  
TransferBurstCount, 506  
TransferComponentSelector, 506  
TransferControlMode, 506  
TransferOperationMode, 506  
TransferPause, 506  
TransferQueueCurrentBlockCount, 506  
TransferQueueMaxBlockCount, 506  
TransferQueueMode, 507  
TransferQueueOverflowCount, 507  
TransferResume, 507  
TransferSelector, 507  
TransferStart, 507  
TransferStatus, 507  
TransferStatusSelector, 507  
TransferStop, 507  
TransferStreamChannel, 508  
TransferTriggerActivation, 508  
TransferTriggerMode, 508  
TransferTriggerSelector, 508  
TransferTriggerSource, 508  
TriggerActivation, 508  
TriggerDelay, 508  
TriggerDivider, 509  
TriggerEventTest, 509  
TriggerMode, 509  
TriggerMultiplier, 509  
TriggerOverlap, 509  
TriggerSelector, 509  
TriggerSoftware, 509  
TriggerSource, 510  
UserOutputSelector, 510  
UserOutputValue, 510  
UserOutputValueAll, 510  
UserOutputValueAllMask, 510  
UserSetDefault, 510  
UserSetFeatureEnable, 511  
UserSetLoad, 511  
UserSetSave, 511  
UserSetSelector, 511  
V3\_3Enable, 511  
WhiteClip, 511  
WhiteClipSelector, 512  
Width, 512

- WidthMax, [512](#)
- Spinnaker::CameraBase
  - ~CameraBase, [515](#)
  - BeginAcquisition, [515](#)
  - CameraBase, [515](#)
  - DeInit, [515](#)
  - DiscoverMaxPacketSize, [516](#)
  - EndAcquisition, [516](#)
  - GetAccessMode, [516](#)
  - GetGuiXml, [516](#)
  - GetNextImage, [517](#)
  - GetNodeMap, [517](#)
  - GetNumDataStreams, [517](#)
  - GetNumImagesInUse, [518](#)
  - GetTLDeviceNodeMap, [518](#)
  - GetTLStreamNodeMap, [518](#)
  - GetUniqueID, [518](#)
  - Init, [519](#)
  - InterfaceImpl, [521](#)
  - IsInitialized, [519](#)
  - IsStreaming, [519](#)
  - IsValid, [520](#)
  - operator=, [520](#)
  - ReadPort, [520](#)
  - RegisterEvent, [520](#), [521](#)
  - UnregisterEvent, [521](#)
  - WritePort, [521](#)
- Spinnaker::CameraList
  - ~CameraList, [523](#)
  - Append, [523](#)
  - CameraList, [523](#)
  - Clear, [523](#)
  - GetByIndex, [524](#)
  - GetBySerial, [524](#)
  - GetSize, [524](#)
  - operator=, [525](#)
  - operator[], [525](#)
  - RemoveByIndex, [525](#)
  - RemoveBySerial, [525](#)
- Spinnaker::ChunkData
  - ~ChunkData, [568](#)
  - ChunkData, [568](#)
  - GetBlackLevel, [568](#)
  - GetCRC, [568](#)
  - GetCounterValue, [568](#)
  - GetEncoderValue, [569](#)
  - GetExposureEndLineStatusAll, [569](#)
  - GetExposureTime, [569](#)
  - GetFrameID, [569](#)
  - GetGain, [569](#)
  - GetHeight, [569](#)
  - GetImage, [570](#)
  - GetInferenceConfidence, [570](#)
  - GetInferenceResult, [570](#)
  - GetLinePitch, [570](#)
  - GetLineStatusAll, [570](#)
  - GetOffsetX, [570](#)
  - GetOffsetY, [571](#)
  - GetPartSelector, [571](#)
  - GetPixelDynamicRangeMax, [571](#)
  - GetPixelDynamicRangeMin, [571](#)
  - GetScan3dAxisMax, [571](#)
  - GetScan3dAxisMin, [571](#)
  - GetScan3dCoordinateOffset, [572](#)
  - GetScan3dCoordinateReferenceValue, [572](#)
  - GetScan3dCoordinateScale, [572](#)
  - GetScan3dInvalidDataValue, [572](#)
  - GetScan3dTransformValue, [572](#)
  - GetScanLineSelector, [572](#)
  - GetSequencerSetActive, [573](#)
  - GetSerialDataLength, [573](#)
  - GetStreamChannelID, [573](#)
  - GetTimerValue, [573](#)
  - GetTimestamp, [573](#)
  - GetTimestampLatchValue, [573](#)
  - GetTransferBlockID, [574](#)
  - GetTransferQueueCurrentBlockCount, [574](#)
  - GetWidth, [574](#)
  - SetChunks, [574](#)
- Spinnaker::DeviceEvent
  - ~DeviceEvent, [616](#)
  - DeviceEvent, [616](#)
  - GetDeviceEventId, [617](#)
  - GetDeviceEventName, [617](#)
  - OnDeviceEvent, [617](#)
  - operator=, [617](#)
- Spinnaker::Event
  - ~Event, [636](#)
  - Event, [636](#)
  - EventProcessor, [637](#)
  - GetEventPayloadData, [636](#)
  - GetEventPayloadDataSize, [636](#)
  - GetEventType, [636](#)
  - IDataStream, [637](#)
  - m\_pEventData, [637](#)
  - operator=, [637](#)
  - SetEventPayload, [637](#)
  - SetEventType, [637](#)
  - Stream, [637](#)
- Spinnaker::Exception
  - ~Exception, [640](#)
  - Exception, [640](#)
  - GetBuildDate, [640](#)
  - GetBuildTime, [640](#)
  - GetError, [640](#)
  - GetErrorMessage, [640](#)
  - GetFileName, [640](#)
  - GetFullErrorMessage, [640](#)
  - GetFunctionName, [640](#)
  - GetLineNumber, [641](#)
  - operator!=, [641](#)
  - operator=, [641](#)
  - operator==, [641](#)
  - what, [641](#)
- Spinnaker::GenApi, [356](#)
- COMMAND\_MAGIC, [372](#)

- GENCP\_COMMAND\_HEADER\_SIZE, 372
- GENCP\_EVENT\_BASIC\_SIZE, 372
- GENCP\_EVENT\_CMD\_ID, 372
- GVCP\_MESSAGE\_TAGS, 371
- IDevFileStream, 371
- IPersistScript, 372
- ODevFileStream, 371
- PersistFeature, 371
- SET\_GUID, 371
- TAG\_EVENT\_CMD, 371
- TAG\_EVENTDATA\_CMD, 371
- U3V\_EVENT\_PREFIX, 372
- Spinnaker::GenApi::AutoLock
  - ~AutoLock, 378
  - AutoLock, 378
- Spinnaker::GenApi::BooleanNode
  - ~BooleanNode, 385
  - BooleanNode, 385
  - GetValue, 385
  - operator=, 385
  - SetReference, 385
  - SetValue, 385
- Spinnaker::GenApi::CChunkAdapter
  - ~CChunkAdapter, 530
  - AttachBuffer, 530
  - AttachNodeMap, 530
  - CChunkAdapter, 530
  - CheckBufferLayout, 530
  - ClearCaches, 530
  - DetachBuffer, 530
  - DetachNodeMap, 531
  - m\_pChunkAdapter, 531
  - UpdateBuffer, 531
- Spinnaker::GenApi::CChunkAdapterDcam
  - ~CChunkAdapterDcam, 532
  - AttachBuffer, 533
  - CChunkAdapterDcam, 532
  - CheckBufferLayout, 533
  - CheckCRC, 533
  - HasCRC, 533
- Spinnaker::GenApi::CChunkAdapterGEV
  - ~CChunkAdapterGEV, 536
  - AttachBuffer, 536
  - CChunkAdapterGEV, 536
  - CheckBufferLayout, 536
- Spinnaker::GenApi::CChunkAdapterGeneric
  - ~CChunkAdapterGeneric, 534
  - AttachBuffer, 534, 535
  - CChunkAdapterGeneric, 534
  - CheckBufferLayout, 535
- Spinnaker::GenApi::CChunkAdapterU3V
  - ~CChunkAdapterU3V, 538
  - AttachBuffer, 538
  - CChunkAdapterU3V, 538
  - CheckBufferLayout, 538
- Spinnaker::GenApi::CChunkPort
  - ~CChunkPort, 540
  - AttachChunk, 540
  - AttachPort, 540
  - CChunkPort, 540
  - CheckChunkID, 540
  - ClearCache, 540
  - DetachChunk, 540
  - DetachPort, 541
  - GetAccessMode, 541
  - GetChunkIDLength, 541
  - GetPrincipalInterfaceType, 541
  - GetSwapEndianness, 541
  - InvalidateNode, 541
  - m\_pChunkPort, 542
  - m\_pPort, 542
  - m\_pPortAdapter, 542
  - Read, 541
  - SetPortImpl, 541
  - UpdateBuffer, 541
  - Write, 541
- Spinnaker::GenApi::CEnumerationTRef
  - ~CEnumerationTRef, 544
  - CEnumerationTRef, 544
  - GetCurrentEntry, 544
  - GetEntry, 544
  - GetValue, 544
  - operator(), 545
  - operator=, 545
  - SetEnumReference, 545
  - SetNumEnums, 545
  - SetReference, 545
  - SetValue, 545
- Spinnaker::GenApi::CEventAdapter
  - ~CEventAdapter, 547
  - AttachNodeMap, 547
  - CEventAdapter, 547
  - DeliverMessage, 547
  - DetachNodeMap, 547
  - m\_pEventAdapter, 547
- Spinnaker::GenApi::CEventAdapter1394
  - ~CEventAdapter1394, 549
  - CEventAdapter1394, 549
  - DeliverEventMessage, 549
  - DeliverMessage, 549
- Spinnaker::GenApi::CEventAdapterGEV
  - ~CEventAdapterGEV, 552
  - CEventAdapterGEV, 552
  - DeliverEventMessage, 552
  - DeliverMessage, 552
- Spinnaker::GenApi::CEventAdapterGeneric
  - ~CEventAdapterGeneric, 550
  - CEventAdapterGeneric, 550
  - DeliverMessage, 551
- Spinnaker::GenApi::CEventAdapterU3V
  - ~CEventAdapterU3V, 554
  - CEventAdapterU3V, 554
  - DeliverEventMessage, 554
  - DeliverMessage, 554
- Spinnaker::GenApi::CEventPort
  - ~CEventPort, 556

- AttachEvent, [556](#)
- AttachNode, [556](#)
- CEventPort, [556](#)
- CheckEventID, [556](#)
- DetachEvent, [556](#)
- DetachNode, [556](#)
- GetAccessMode, [557](#)
- GetEventIDLength, [557](#)
- GetPrincipalInterfaceType, [557](#)
- GetSwapEndianness, [557](#)
- InvalidateNode, [557](#)
- m\_pEventPort, [557](#)
- m\_pNode, [557](#)
- m\_pPortAdapter, [557](#)
- Read, [557](#)
- SetPortImpl, [557](#)
- Write, [557](#)
- Spinnaker::GenApi::CFeatureBag
  - ~CFeatureBag, [559](#)
  - CFeatureBag, [559](#)
  - GetFeatureBagHandle, [559](#)
  - LoadFromBag, [559](#)
  - operator==, [559](#)
  - PersistFeature, [559](#)
  - SetInfo, [559](#)
  - StoreToBag, [559](#)
- Spinnaker::GenApi::CFloatPtr
  - CFloatPtr, [561](#)
  - GetEnumAlias, [561](#)
  - GetIntAlias, [561](#)
  - operator=, [561](#)
- Spinnaker::GenApi::CGeneric\_XMLLoaderParams
  - \_Initialize, [562](#)
- Spinnaker::GenApi::CLock
  - ~CLock, [576](#)
  - CLock, [576](#)
  - Lock, [576](#)
  - m\_bOwnLock, [576](#)
  - m\_lock, [576](#)
  - NodeMap, [576](#)
  - TryLock, [576](#)
  - Unlock, [576](#)
- Spinnaker::GenApi::CLockEx
  - m\_lockEx, [579](#)
- Spinnaker::GenApi::CNodeCallback
  - ~CNodeCallback, [581](#)
  - CNodeCallback, [581](#)
  - Destroy, [581](#)
  - GetCallbackType, [581](#)
  - GetNode, [581](#)
  - m\_CallbackType, [581](#)
  - m\_pNode, [581](#)
  - operator(), [581](#)
- Spinnaker::GenApi::CNodeMapFactory
  - ~CNodeMapFactory, [584](#)
  - AddInjectionData, [586](#)
  - ApplyStyleSheet, [586](#)
  - CNodeMapFactory, [584](#), [585](#)
  - ClearCache, [586](#)
  - CreateEmptyNodeMap, [586](#)
  - CreateNodeDataFromNodeMap, [586](#)
  - CreateNodeMap, [587](#)
  - ExtractSubtree, [587](#)
  - GetNodeStatistics, [587](#)
  - GetSupportedSchemaVersions, [587](#)
  - IsCameraDescriptionFileDataReleased, [587](#)
  - IsEmpty, [588](#)
  - IsLoaded, [588](#)
  - IsPreprocessed, [588](#)
  - LoadAndInject, [588](#)
  - operator=, [588](#)
  - Preprocess, [588](#)
  - ReleaseCameraDescriptionFileData, [588](#)
  - ToString, [589](#)
  - ToXml, [589](#)
- Spinnaker::GenApi::CNodeMapFactory::NodeStatistics←\_t
  - NumLinks, [784](#)
  - NumNodes, [784](#)
  - NumProperties, [784](#)
  - NumStrings, [784](#)
- Spinnaker::GenApi::CNodeMapRef
  - CNodeMapRef, [590](#), [591](#)
  - operator=, [591](#)
- Spinnaker::GenApi::CNodeMapRefT
  - \_ClearXMLCache, [593](#)
  - \_Connect, [593](#)
  - \_GetDeviceName, [594](#)
  - \_GetNode, [594](#)
  - \_GetNodes, [594](#)
  - \_GetSupportedSchemaVersions, [594](#)
  - \_InvalidateNodes, [594](#)
  - \_LoadXMLFromFile, [594](#)
  - \_LoadXMLFromFileInject, [594](#)
  - \_LoadXMLFromString, [594](#)
  - \_LoadXMLFromStringInject, [594](#)
  - \_LoadXMLFromZIPData, [595](#)
  - \_LoadXMLFromZIPFile, [595](#)
  - \_Poll, [595](#)
  - \_Ptr, [595](#)
- Spinnaker::GenApi::CPointer
  - ~CPointer, [600](#)
  - CPointer, [600](#)
  - IsValid, [600](#)
  - m\_pT, [602](#)
  - operator bool, [600](#)
  - operator T \*, [600](#)
  - operator!=, [600](#), [601](#)
  - operator\*, [601](#)
  - operator(), [601](#)
  - operator->, [601](#)
  - operator=, [601](#)
  - operator==, [601](#)
- Spinnaker::GenApi::CPortImpl
  - ~CPortImpl, [603](#)
  - CPortImpl, [603](#)

- GetAccessMode, 603
- GetSwapEndianness, 603
- InvalidateNode, 604
- m\_ptrPort, 604
- Read, 604
- Replay, 604
- SetPortImpl, 604
- Write, 604
- Spinnaker::GenApi::CPortWriteList
  - ~CPortWriteList, 606
  - CPortWriteList, 606
  - GetCookie, 606
  - GetPortWriteListHandle, 606
  - m\_pWriteList, 607
  - Replay, 606
  - SetCookie, 606
  - Write, 606
- Spinnaker::GenApi::CRegisterPortImpl
  - ~CRegisterPortImpl, 608
  - CRegisterPortImpl, 608
  - GetAccessMode, 608
  - Read, 608
  - ReadRegister, 609
  - SetPortImpl, 609
  - Write, 609
  - WriteRegister, 609
- Spinnaker::GenApi::CSelectorSet
  - ~CSelectorSet, 611
  - CSelectorSet, 610
  - GetSelectorList, 611
  - IsEmpty, 611
  - Restore, 611
  - SetFirst, 611
  - SetNext, 611
  - ToString, 611
- Spinnaker::GenApi::CTestPortStruct
  - CTestPortStruct, 613
  - GetAccessMode, 613
  - GetNumReads, 613
  - GetNumWrites, 613
  - GetPrincipalInterfaceType, 613
  - m\_BaseAddress, 614
  - m\_NumReads, 614
  - m\_NumWrites, 614
  - MemSet, 614
  - Read, 614
  - ResetStatistics, 614
  - Write, 614
- Spinnaker::GenApi::CategoryNode
  - ~CategoryNode, 528
  - CategoryNode, 528
  - GetFeatures, 528
  - SetReference, 528
- Spinnaker::GenApi::CommandNode
  - ~CommandNode, 597
  - CommandNode, 597
  - Execute, 597
  - IsDone, 597
  - operator(), 597
  - SetReference, 597
- Spinnaker::GenApi::Counter
  - Counter, 598
  - GetValue, 598
  - IsZero, 598
  - operator unsigned int, 598
  - operator++, 598
  - operator--, 598
- Spinnaker::GenApi::EAccessModeClass
  - FromString, 619
  - ToString, 619
- Spinnaker::GenApi::ECachingModeClass
  - FromString, 620
  - ToString, 620
- Spinnaker::GenApi::EDisplayNotationClass
  - FromString, 621
  - ToString, 621
- Spinnaker::GenApi::EEndiannessClass
  - FromString, 622
  - ToString, 622
- Spinnaker::GenApi::EGenApiSchemaVersionClass
  - FromString, 623
  - ToString, 623
- Spinnaker::GenApi::EInputDirectionClass
  - FromString, 623
  - ToString, 623, 624
- Spinnaker::GenApi::ENameSpaceClass
  - FromString, 624
  - ToString, 624
- Spinnaker::GenApi::ERepresentationClass
  - FromString, 632
  - ToString, 632
- Spinnaker::GenApi::ESignClass
  - FromString, 633
  - ToString, 633
- Spinnaker::GenApi::ESlopeClass
  - FromString, 634
  - ToString, 634
- Spinnaker::GenApi::EStandardNameSpaceClass
  - FromString, 635
  - ToString, 635
- Spinnaker::GenApi::EVisibilityClass
  - FromString, 638
  - ToString, 638
- Spinnaker::GenApi::EYesNoClass
  - FromString, 642
  - ToString, 642
- Spinnaker::GenApi::EnumEntryNode
  - ~EnumEntryNode, 627
  - EnumEntryNode, 627
  - GetNumericValue, 627
  - GetSymbolic, 627
  - GetValue, 627
  - IsSelfClearing, 627
  - SetReference, 627
- Spinnaker::GenApi::EnumNode
  - ~EnumNode, 630

- EnumNode, [630](#)
- GetCurrentEntry, [630](#)
- GetEntries, [630](#)
- GetEntry, [630](#)
- GetEntryByName, [630](#)
- GetIntValue, [630](#)
- GetSymbolics, [631](#)
- m\_pEnumeration, [631](#)
- operator\*, [631](#)
- operator=, [631](#)
- SetIntValue, [631](#)
- SetReference, [631](#)
- Spinnaker::GenApi::FileProtocolAdapter
  - ~FileProtocolAdapter, [643](#)
  - attach, [643](#)
  - closeFile, [643](#)
  - deleteFile, [643](#)
  - FileProtocolAdapter, [643](#)
  - getBufSize, [644](#)
  - openFile, [644](#)
  - read, [644](#)
  - write, [644](#)
- Spinnaker::GenApi::FloatNode
  - ~FloatNode, [647](#)
  - FloatNode, [647](#)
  - GetDisplayNotation, [647](#)
  - GetDisplayPrecision, [647](#)
  - GetEnumAlias, [647](#)
  - GetInc, [647](#)
  - GetIncMode, [648](#)
  - GetIntAlias, [648](#)
  - GetListOfValidValues, [648](#)
  - GetMax, [648](#)
  - GetMin, [648](#)
  - GetRepresentation, [648](#)
  - GetUnit, [648](#)
  - GetValue, [648](#)
  - HasInc, [649](#)
  - ImposeMax, [649](#)
  - ImposeMin, [649](#)
  - operator\*, [649](#)
  - operator(), [649](#)
  - operator=, [649](#)
  - SetReference, [649](#)
  - SetValue, [649](#)
- Spinnaker::GenApi::FloatRegNode
  - ~FloatRegNode, [651](#)
  - FloatRegNode, [651](#)
  - SetReference, [652](#)
- Spinnaker::GenApi::Function\_NodeCallback
  - Destroy, [653](#)
  - Function\_NodeCallback, [653](#)
  - operator(), [653](#)
- Spinnaker::GenApi::IDevFileStreamBase
  - close, [681](#)
  - filebuf\_type, [681](#)
  - ios\_type, [681](#)
  - is\_open, [681](#)
  - istream\_type, [681](#)
  - open, [681](#)
  - rdbuf, [681](#)
- Spinnaker::GenApi::IDevFileStreamBuf
  - ~IDevFileStreamBuf, [683](#)
  - close, [683](#)
  - IDevFileStreamBuf, [683](#)
  - is\_open, [683](#)
  - open, [683](#)
  - pbackfail, [683](#)
  - underflow, [683](#)
- Spinnaker::GenApi::IntRegNode
  - ~IntRegNode, [751](#)
  - IntRegNode, [751](#)
  - SetReference, [752](#)
- Spinnaker::GenApi::IntegerNode
  - ~IntegerNode, [737](#)
  - GetFloatAlias, [737](#)
  - GetInc, [737](#)
  - GetIncMode, [737](#)
  - GetListOfValidValues, [737](#)
  - GetMax, [737](#)
  - GetMin, [738](#)
  - GetRepresentation, [738](#)
  - GetUnit, [738](#)
  - GetValue, [738](#)
  - ImposeMax, [738](#)
  - ImposeMin, [738](#)
  - IntegerNode, [737](#)
  - operator\*, [738](#)
  - operator(), [738](#)
  - operator=, [739](#)
  - SetReference, [739](#)
  - SetValue, [739](#)
- Spinnaker::GenApi::Member\_NodeCallback
  - Destroy, [769](#)
  - Member\_NodeCallback, [769](#)
  - operator(), [769](#)
  - PMEMBERFUNC, [768](#)
- Spinnaker::GenApi::Node
  - ~Node, [773](#)
  - DeregisterCallback, [773](#)
  - GetAccessMode, [773](#)
  - GetAlias, [773](#)
  - GetCachingMode, [774](#)
  - GetCastAlias, [774](#)
  - GetChildren, [774](#)
  - GetDescription, [774](#)
  - GetDeviceName, [774](#)
  - GetDisplayName, [774](#)
  - GetDocuURL, [774](#)
  - GetEventID, [774](#)
  - GetName, [774](#)
  - GetNameSpace, [775](#)
  - GetNodeHandle, [775](#)
  - GetNodeMap, [775](#)
  - GetParents, [775](#)
  - GetPollingTime, [775](#)

- GetPrincipalInterfaceType, 775
- GetProperty, 775
- GetPropertyNames, 775
- GetSelectedFeatures, 775
- GetSelectingFeatures, 776
- GetToolTip, 776
- GetVisibility, 776
- ImposeAccessMode, 776
- ImposeVisibility, 776
- InvalidateNode, 776
- IsAccessModeCacheable, 776
- IsCachable, 776
- IsDeprecated, 776
- IsFeature, 776
- IsSelector, 777
- IsStreamable, 777
- m\_Callbacks, 777
- m\_pNodeData, 777
- m\_pNodeMap, 778
- Node, 773
- operator!=, 777
- operator==, 777
- RegisterCallback, 777
- SetNodeHandle, 777
- SetNodeMap, 777
- SetReference, 777
- Spinnaker::GenApi::NodeMap
  - \_Ptr, 784
  - ~NodeMap, 780
  - ClearXMLCache, 780
  - Connect, 780, 781
  - Destroy, 781
  - GetDeviceName, 781
  - GetDeviceVersion, 781
  - GetGenApiVersion, 781
  - GetLock, 781
  - GetModelName, 781
  - GetNode, 781
  - GetNodeMapHandle, 781
  - GetNodes, 781
  - GetNumNodes, 781
  - GetProductGuid, 782
  - GetSchemaVersion, 782
  - GetStandardNameSpace, 782
  - GetSupportedSchemaVersions, 782
  - GetToolTip, 782
  - GetVendorName, 782
  - GetVersionGuid, 783
  - InvalidateNodes, 783
  - LoadXMLFromFile, 783
  - LoadXMLFromFileInject, 783
  - LoadXMLFromString, 783
  - LoadXMLFromStringInject, 783
  - LoadXMLFromZIPData, 783
  - LoadXMLFromZIPFile, 783
  - NodeMap, 780
  - Poll, 783
- Spinnaker::GenApi::ODevFileStreamBase
  - close, 786
  - filebuf\_type, 786
  - ios\_type, 786
  - is\_open, 786
  - open, 786
  - ostream\_type, 786
  - rdbuf, 786
- Spinnaker::GenApi::ODevFileStreamBuf
  - ~ODevFileStreamBuf, 788
  - close, 788
  - is\_open, 788
  - ODevFileStreamBuf, 788
  - open, 788
  - overflow, 788
  - sync, 788
  - xspn, 788
- Spinnaker::GenApi::PortNode
  - ~PortNode, 792
  - CacheChunkData, 792
  - GetChunkID, 792
  - GetPortHandle, 792
  - GetSwapEndianness, 792
  - PortNode, 792
  - Read, 792
  - Replay, 792
  - SetPortImpl, 793
  - SetReference, 793
  - StartRecording, 793
  - StopRecording, 793
  - Write, 793
- Spinnaker::GenApi::PortRecorder
  - ~PortRecorder, 796
  - GetAccessMode, 796
  - PortRecorder, 796
  - SetReference, 796
  - StartRecording, 796
  - StopRecording, 796
- Spinnaker::GenApi::PortReplay
  - ~PortReplay, 798
  - GetPortReplayHandle, 798
  - PortReplay, 798
  - Replay, 798
  - SetReference, 798
- Spinnaker::GenApi::RegisterNode
  - ~RegisterNode, 802
  - Get, 802
  - GetAddress, 802
  - GetLength, 802
  - RegisterNode, 802
  - Set, 802
  - SetReference, 802
- Spinnaker::GenApi::StringNode
  - ~StringNode, 811
  - GetMaxLength, 811
  - GetValue, 811
  - operator\*, 812
  - operator(), 812
  - operator=, 812



- SetReference, [812](#)
- SetValue, [812](#)
- StringNode, [811](#)
- Spinnaker::GenApi::StringRegNode
  - ~StringRegNode, [814](#)
  - SetReference, [815](#)
  - StringRegNode, [814](#)
- Spinnaker::GenApi::ValueNode
  - ~ValueNode, [848](#)
  - FromString, [848](#)
  - GetNode, [849](#)
  - IsValueCacheValid, [849](#)
  - SetReference, [849](#)
  - ToString, [849](#)
  - ValueNode, [848](#)
- Spinnaker::GenApi::double\_autovector\_t
  - \_pCount, [619](#)
  - \_pv, [619](#)
  - ~double\_autovector\_t, [618](#)
  - double\_autovector\_t, [618](#)
  - operator delete, [618](#)
  - operator new, [618](#)
  - operator=, [618](#)
  - operator[], [618](#)
  - size, [619](#)
- Spinnaker::GenApi::int64\_autovector\_t
  - \_pCount, [734](#)
  - \_pv, [734](#)
  - ~int64\_autovector\_t, [734](#)
  - int64\_autovector\_t, [734](#)
  - operator delete, [734](#)
  - operator new, [734](#)
  - operator=, [734](#)
  - operator[], [734](#)
  - size, [734](#)
- Spinnaker::GenICam, [372](#)
  - getline, [374](#)
  - ThrowBadAlloc, [374](#)
- Spinnaker::GenICam::AutoLock
  - ~AutoLock, [379](#)
  - AutoLock, [379](#)
- Spinnaker::GenICam::CGlobalLock
  - ~CGlobalLock, [563](#)
  - CGlobalLock, [563](#)
  - IsValid, [563](#)
  - Lock, [563](#)
  - m\_DebugCount, [564](#)
  - TryLock, [564](#)
  - Unlock, [564](#)
- Spinnaker::GenICam::CGlobalLockUnlocker
  - ~CGlobalLockUnlocker, [565](#)
  - CGlobalLockUnlocker, [565](#)
  - m\_Lock, [565](#)
  - m\_enabled, [565](#)
  - UnlockEarly, [565](#)
- Spinnaker::GenICam::CLock
  - ~CLock, [577](#)
  - CLock, [577](#)
- Lock, [578](#)
- TryLock, [578](#)
- Unlock, [578](#)
- Spinnaker::GenICam::LockableObject
  - GetLock, [761](#)
  - Lock, [761](#)
  - m\_Lock, [761](#)
- Spinnaker::GenICam::LockableObject::Lock
  - ~Lock, [759](#)
  - Lock, [759](#)
- Spinnaker::GenICam::Version\_t
  - Major, [850](#)
  - Minor, [850](#)
  - SubMinor, [850](#)
- Spinnaker::GenICam::gcstring
  - \_npos, [655](#)
  - ~gcstring, [655](#)
  - append, [655](#)
  - assign, [655](#)
  - c\_str, [655](#)
  - compare, [655](#)
  - empty, [655](#)
  - find, [655](#), [656](#)
  - find\_first\_not\_of, [656](#)
  - find\_first\_of, [656](#)
  - gcstring, [655](#)
  - length, [656](#)
  - max\_size, [656](#)
  - npos, [657](#)
  - operator const char \*, [656](#)
  - operator delete, [656](#)
  - operator new, [656](#)
  - operator!=, [656](#)
  - operator<, [656](#)
  - operator>, [657](#)
  - operator+, [657](#)
  - operator+=", [656](#)
  - operator=, [656](#)
  - operator==, [656](#), [657](#)
  - resize, [657](#)
  - size, [657](#)
  - substr, [657](#)
  - swap, [657](#)
- Spinnaker::IArrivalEvent
  - ~IArrivalEvent, [667](#)
  - IArrivalEvent, [667](#)
  - OnDeviceArrival, [667](#)
  - operator=, [667](#)
- Spinnaker::ICameraBase
  - ~ICameraBase, [669](#)
  - BeginAcquisition, [669](#)
  - CameraInternal, [672](#)
  - DelInit, [669](#)
  - DiscoverMaxPacketSize, [669](#)
  - EndAcquisition, [669](#)
  - GetAccessMode, [669](#)
  - GetGuiXml, [670](#)
  - GetNextImage, [670](#)



- GetNodeMap, [670](#)
- GetNumDataStreams, [670](#)
- GetNumImagesInUse, [670](#)
- GetTLDeviceNodeMap, [670](#)
- GetTLStreamNodeMap, [670](#)
- GetUniqueID, [670](#)
- ICameraBase, [669](#)
- Init, [670](#)
- InterfaceImpl, [672](#)
- IsInitialized, [670](#)
- IsStreaming, [671](#)
- IsValid, [671](#)
- m\_pCameraBaseData, [672](#)
- operator=, [671](#)
- ReadPort, [671](#)
- RegisterEvent, [671](#)
- TLDevice, [672](#)
- TLStream, [672](#)
- UnregisterEvent, [671](#)
- WritePort, [671](#)
- Spinnaker::ICameraList
  - ~ICameraList, [673](#)
  - Append, [673](#)
  - CameraListImpl, [674](#)
  - Clear, [673](#)
  - GetByIndex, [674](#)
  - GetBySerial, [674](#)
  - GetSize, [674](#)
  - ICameraList, [673](#)
  - InterfaceImpl, [674](#)
  - m\_pCameraListData, [674](#)
  - operator=, [674](#)
  - operator[], [674](#)
  - RemoveByIndex, [674](#)
  - RemoveBySerial, [674](#)
- Spinnaker::IChunkData
  - ~IChunkData, [676](#)
  - GetBlackLevel, [676](#)
  - GetCRC, [676](#)
  - GetCounterValue, [676](#)
  - GetEncoderValue, [676](#)
  - GetExposureEndLineStatusAll, [676](#)
  - GetExposureTime, [676](#)
  - GetFrameID, [677](#)
  - GetGain, [677](#)
  - GetHeight, [677](#)
  - GetImage, [677](#)
  - GetInferenceConfidence, [677](#)
  - GetInferenceResult, [677](#)
  - GetLinePitch, [677](#)
  - GetLineStatusAll, [677](#)
  - GetOffsetX, [677](#)
  - GetOffsetY, [677](#)
  - GetPartSelector, [678](#)
  - GetPixelDynamicRangeMax, [678](#)
  - GetPixelDynamicRangeMin, [678](#)
  - GetScan3dAxisMax, [678](#)
  - GetScan3dAxisMin, [678](#)
  - GetScan3dCoordinateOffset, [678](#)
  - GetScan3dCoordinateReferenceValue, [678](#)
  - GetScan3dCoordinateScale, [678](#)
  - GetScan3dInvalidDataValue, [678](#)
  - GetScan3dTransformValue, [678](#)
  - GetScanLineSelector, [679](#)
  - GetSequencerSetActive, [679](#)
  - GetSerialDataLength, [679](#)
  - GetStreamChannelID, [679](#)
  - GetTimerValue, [679](#)
  - GetTimestamp, [679](#)
  - GetTimestampLatchValue, [679](#)
  - GetTransferBlockID, [679](#)
  - GetTransferQueueCurrentBlockCount, [679](#)
  - GetWidth, [679](#)
  - IChunkData, [676](#)
  - SetChunks, [680](#)
- Spinnaker::IDeviceEvent
  - ~IDeviceEvent, [684](#)
  - GetDeviceEventId, [684](#)
  - GetDeviceEventName, [684](#)
  - IDeviceEvent, [684](#)
  - OnDeviceEvent, [684](#)
  - operator=, [685](#)
- Spinnaker::IImage
  - ~IImage, [687](#)
  - CalculateStatistics, [687](#)
  - CheckCRC, [687](#)
  - Convert, [687](#)
  - DeepCopy, [687](#)
  - ExtractPolarization, [687](#)
  - GetBitsPerPixel, [687](#)
  - GetBufferSize, [687](#)
  - GetChunkData, [687](#)
  - GetChunkLayoutId, [687](#)
  - GetColorProcessing, [688](#)
  - GetData, [688](#)
  - GetFrameID, [688](#)
  - GetHeight, [688](#)
  - GetID, [688](#)
  - GetImageSize, [688](#)
  - GetImageStatus, [688](#)
  - GetNumChannels, [688](#)
  - GetPayloadType, [688](#)
  - GetPixelFormat, [688](#)
  - GetPixelFormatIntType, [689](#)
  - GetPixelFormatName, [689](#)
  - GetPolarizationAlgorithm, [689](#)
  - GetPolarizationValues, [689](#)
  - GetPrivateData, [689](#)
  - GetStride, [689](#)
  - GetTLPayloadType, [689](#)
  - GetTLPixelFormat, [689](#)
  - GetTLPixelFormatNamespace, [689](#)
  - GetTimeStamp, [689](#)
  - GetValidPayloadSize, [690](#)
  - GetWidth, [690](#)
  - GetXOffset, [690](#)

- GetXPadding, 690
- GetYOffset, 690
- GetYPadding, 690
- HasCRC, 690
- Image, 687
- IsInUse, 690
- IsIncomplete, 690
- Release, 690
- ResetImage, 691
- Save, 691
- Spinnaker::IImageEvent
  - ~IImageEvent, 693
  - IImageEvent, 693
  - OnImageEvent, 693
  - operator=, 693
- Spinnaker::IImageStatistics
  - ~IImageStatistics, 694
  - DisableAll, 694
  - EnableAll, 694
  - EnableGreyOnly, 694
  - EnableHSLOnly, 695
  - EnableRGBOnly, 695
  - GetChannelStatus, 695
  - GetHistogram, 695
  - GetMean, 695
  - GetNumPixelValues, 695
  - GetPixelValueRange, 695
  - GetRange, 695
  - GetStatistics, 695
  - IImageStatistics, 694
  - SetChannelStatus, 696
- Spinnaker::IInterface
  - ~IInterface, 697
  - GetCameras, 697
  - GetTLNodeMap, 697
  - IInterface, 697
  - InterfaceInternal, 698
  - IsInUse, 698
  - m\_pInterfaceData, 698
  - operator=, 698
  - RegisterEvent, 698
  - SendActionCommand, 698
  - TLInterface, 698
  - UnregisterEvent, 698
  - UpdateCameras, 698
- Spinnaker::IInterfaceEvent
  - ~IInterfaceEvent, 700
  - IInterfaceEvent, 700
  - OnDeviceArrival, 700
  - OnDeviceRemoval, 700
  - operator=, 700
- Spinnaker::IInterfaceList
  - ~IInterfaceList, 701
  - Clear, 701
  - GetByIndex, 701
  - GetSize, 701
  - IInterfaceList, 701
  - m\_pInterfaceListData, 702
  - operator=, 701
  - operator[], 702
- Spinnaker::ILoggingEvent
  - ~ILoggingEvent, 703
  - ILoggingEvent, 703
  - OnLogEvent, 703
  - operator=, 703
- Spinnaker::IRemovalEvent
  - ~IRemovalEvent, 753
  - IRemovalEvent, 753
  - OnDeviceRemoval, 753
  - operator=, 753
- Spinnaker::ISystem
  - ~ISystem, 754
  - GetCameras, 754
  - GetInterfaces, 754
  - GetLibraryVersion, 754
  - GetLoggingEventPriorityLevel, 755
  - ISystem, 754
  - IsInUse, 755
  - operator=, 755
  - RegisterInterfaceEvent, 755
  - RegisterLoggingEvent, 755
  - ReleaseInstance, 755
  - SendActionCommand, 755
  - SetLoggingEventPriorityLevel, 755
  - UnregisterAllLoggingEvent, 755
  - UnregisterInterfaceEvent, 755
  - UnregisterLoggingEvent, 756
  - UpdateCameras, 756
- Spinnaker::Image
  - ~Image, 707
  - CalculateStatistics, 708
  - CheckCRC, 708
  - Convert, 708, 709
  - Create, 709
  - CreateShared, 709
  - DeepCopy, 709, 710
  - ExtractPolarization, 710
  - GetBitsPerPixel, 710
  - GetBufferSize, 710
  - GetChunkData, 710
  - GetChunkLayoutId, 711
  - GetColorProcessing, 711
  - GetData, 711
  - GetDefaultColorProcessing, 711
  - GetFrameID, 712
  - GetHeatMapColorGradient, 712
  - GetHeatMapRange, 712
  - GetHeight, 713
  - GetID, 713
  - GetImageSize, 713
  - GetImageStatus, 713
  - GetImageStatusDescription, 713
  - GetNumChannels, 714
  - GetPayloadType, 714
  - GetPixelFormat, 714
  - GetPixelFormatIntType, 714

- GetPixelFormatName, [715](#)
- GetPolarizationAlgorithm, [715](#)
- GetPolarizationValues, [715](#)
- GetPrivateData, [715](#)
- GetStride, [716](#)
- GetTLPayloadType, [716](#)
- GetTLPixelFormat, [716](#)
- GetTLPixelFormatNamespace, [717](#)
- GetTimeStamp, [716](#)
- GetValidPayloadSize, [717](#)
- GetWidth, [717](#)
- GetXOffset, [718](#)
- GetXPadding, [718](#)
- GetYOffset, [718](#)
- GetYPadding, [718](#)
- HasCRC, [719](#)
- IDataStream, [723](#)
- Image, [707](#), [708](#)
- ImageConverter, [723](#)
- ImageFiler, [723](#)
- ImageStatsCalculator, [723](#)
- IsInUse, [719](#)
- IsIncomplete, [719](#)
- m\_pImageData, [723](#)
- Release, [719](#)
- ResetImage, [719](#), [720](#)
- Save, [720–722](#)
- SetDefaultColorProcessing, [722](#)
- SetHeatMapColorGradient, [722](#)
- SetHeatMapRange, [723](#)
- Stream, [723](#)
- Spinnaker::ImageEvent
  - ~ImageEvent, [725](#)
  - ImageEvent, [725](#)
  - OnImageEvent, [725](#)
  - operator=, [725](#)
- Spinnaker::ImagePtr
  - ~ImagePtr, [727](#)
  - ImagePtr, [727](#)
  - operator=, [727](#)
- Spinnaker::ImageStatistics
  - ~ImageStatistics, [729](#)
  - DisableAll, [730](#)
  - EnableAll, [730](#)
  - EnableGreyOnly, [730](#)
  - EnableHSLOnly, [730](#)
  - EnableRGBOnly, [730](#)
  - GetChannelStatus, [730](#)
  - GetHistogram, [731](#)
  - GetMean, [731](#)
  - GetNumPixelValues, [731](#)
  - GetPixelValueRange, [731](#)
  - GetRange, [732](#)
  - GetStatistics, [732](#)
  - ImageStatistics, [729](#)
  - ImageStatsCalculator, [733](#)
  - operator=, [732](#)
  - SetChannelStatus, [733](#)
- Spinnaker::Interface
  - ~Interface, [741](#)
  - GetCameras, [741](#)
  - GetTLNodeMap, [741](#)
  - InterfaceInternal, [743](#)
  - IsInUse, [741](#)
  - RegisterEvent, [741](#)
  - SendActionCommand, [742](#)
  - UnregisterEvent, [742](#)
  - UpdateCameras, [742](#)
- Spinnaker::InterfaceEvent
  - ~InterfaceEvent, [745](#)
  - InterfaceEvent, [745](#)
  - OnDeviceArrival, [745](#)
  - OnDeviceRemoval, [745](#)
  - operator=, [745](#)
- Spinnaker::InterfaceList
  - ~InterfaceList, [747](#)
  - Clear, [747](#)
  - GetByIndex, [747](#)
  - GetSize, [747](#)
  - InterfaceList, [747](#)
  - operator=, [747](#)
  - operator[], [748](#)
  - SystemImpl, [748](#)
- Spinnaker::InterfacePtr
  - ~InterfacePtr, [749](#)
  - InterfacePtr, [749](#)
  - operator=, [749](#)
- Spinnaker::JPEGOption
  - JPEGOption, [756](#)
  - progressive, [756](#)
  - quality, [756](#)
  - reserved, [757](#)
- Spinnaker::JPG2Option
  - JPG2Option, [757](#)
  - quality, [757](#)
  - reserved, [757](#)
- Spinnaker::LibraryVersion
  - build, [758](#)
  - major, [758](#)
  - minor, [758](#)
  - type, [758](#)
- Spinnaker::LoggingEvent
  - ~LoggingEvent, [762](#)
  - LoggingEvent, [762](#)
  - OnLogEvent, [763](#)
  - operator=, [763](#)
- Spinnaker::LoggingEventData
  - ~LoggingEventData, [764](#)
  - GetCategoryName, [764](#)
  - GetLogMessage, [764](#)
  - GetNDC, [764](#)
  - GetPriority, [765](#)
  - GetPriorityName, [765](#)
  - GetThreadName, [765](#)
  - GetTimestamp, [765](#)
  - LoggingEventData, [764](#)

- SystemImpl, 765
- Spinnaker::LoggingEventDataPtr
  - ~LoggingEventDataPtr, 767
  - LoggingEventDataPtr, 767
  - operator=, 767
- Spinnaker::PGMOption
  - binaryFile, 789
  - PGMOption, 789
  - reserved, 789
- Spinnaker::PNGOption
  - compressionLevel, 790
  - interlaced, 790
  - PNGOption, 790
  - reserved, 790
- Spinnaker::PPMOption
  - binaryFile, 799
  - PPMOption, 799
  - reserved, 799
- Spinnaker::RemovalEvent
  - ~RemovalEvent, 804
  - OnDeviceRemoval, 804
  - operator=, 804
  - RemovalEvent, 804
- Spinnaker::System
  - ~System, 817
  - GetCameras, 817
  - GetInstance, 817
  - GetInterfaces, 818
  - GetLibraryVersion, 818
  - GetLoggingEventPriorityLevel, 818
  - IsInUse, 818
  - RegisterInterfaceEvent, 819
  - RegisterLoggingEvent, 819
  - ReleaseInstance, 819
  - SendActionCommand, 819
  - SetLoggingEventPriorityLevel, 820
  - System, 817
  - UnregisterAllLoggingEvent, 820
  - UnregisterInterfaceEvent, 821
  - UnregisterLoggingEvent, 821
  - UpdateCameras, 821
- Spinnaker::SystemPtr
  - ~SystemPtr, 823
  - SystemPtr, 823
- Spinnaker::TIFFOption
  - ADOBE\_DEFLATE, 824
  - CCITTFAX3, 824
  - CCITTFAX4, 824
  - compression, 824
  - CompressionMethod, 824
  - DEFLATE, 824
  - JPEG, 824
  - LZW, 824
  - NONE, 824
  - PACKBITS, 824
  - reserved, 824
  - TIFFOption, 824
- Spinnaker::TransportLayerDevice
  - ~TransportLayerDevice, 827
  - CameraBase, 827
  - CameraInternal, 827
  - DeviceAccessStatus, 827
  - DeviceCurrentSpeed, 827
  - DeviceDisplayName, 827
  - DeviceDriverVersion, 827
  - DeviceEndiannessMechanism, 827
  - DeviceID, 827
  - DeviceInstanceId, 828
  - DevicesUpdater, 828
  - DeviceLinkSpeed, 828
  - DeviceModelName, 828
  - DeviceMulticastMonitorMode, 828
  - DeviceSerialNumber, 828
  - DeviceType, 828
  - DeviceU3VProtocol, 828
  - DeviceUserID, 829
  - DeviceVendorName, 829
  - DeviceVersion, 829
  - GUIXMLLocation, 831
  - GUIXMLPath, 831
  - GenICamXMLLocation, 829
  - GenICamXMLPath, 829
  - GevCCP, 829
  - GevDeviceDiscoverMaximumPacketSize, 829
  - GevDeviceGateway, 829
  - GevDeviceIPAddress, 830
  - GevDevicesWrongSubnet, 830
  - GevDeviceMACAddress, 830
  - GevDeviceMaximumPacketSize, 830
  - GevDeviceMaximumRetryCount, 830
  - GevDeviceModelsBigEndian, 830
  - GevDevicePort, 830
  - GevDeviceReadAndWriteTimeout, 831
  - GevDeviceSubnetMask, 831
  - GevVersionMajor, 831
  - GevVersionMinor, 831
  - ICameraBase, 827
  - TransportLayerDevice, 827
- Spinnaker::TransportLayerInterface
  - ~TransportLayerInterface, 834
  - ActionCommand, 834
  - AutoForceIP, 834
  - DeviceAccessStatus, 834
  - DeviceCount, 834
  - DeviceID, 834
  - DeviceModelName, 835
  - DeviceSelector, 835
  - DeviceUnlock, 835
  - DeviceUpdateList, 835
  - DeviceVendorName, 835
  - GevActionDeviceKey, 835
  - GevActionGroupKey, 835
  - GevActionGroupMask, 836
  - GevActionTime, 836
  - GevDeviceIPAddress, 836
  - GevDeviceMACAddress, 836

- GevDeviceSubnetMask, [836](#)
- GevInterfaceGateway, [836](#)
- GevInterfaceIPAddress, [836](#)
- GevInterfaceMACAddress, [837](#)
- GevInterfaceSubnetMask, [837](#)
- HostAdapterDriverVersion, [837](#)
- HostAdapterName, [837](#)
- HostAdapterVendor, [837](#)
- IInterface, [834](#)
- IncompatibleDeviceCount, [837](#)
- IncompatibleDeviceID, [837](#)
- IncompatibleDeviceModelName, [837](#)
- IncompatibleDeviceSelector, [838](#)
- IncompatibleDeviceVendorName, [838](#)
- IncompatibleGevDeviceIPAddress, [838](#)
- IncompatibleGevDeviceMACAddress, [838](#)
- IncompatibleGevDeviceSubnetMask, [838](#)
- Interface, [834](#)
- InterfaceDisplayName, [838](#)
- InterfaceID, [838](#)
- InterfaceInternal, [834](#)
- InterfaceType, [839](#)
- POEStatus, [839](#)
- TransportLayerInterface, [834](#)
- Spinnaker::TransportLayerStream
  - ~TransportLayerStream, [841](#)
  - CameraBase, [841](#)
  - CameraInternal, [841](#)
  - GevFailedPacketCount, [841](#)
  - GevMaximumNumberResendBuffers, [841](#)
  - GevMaximumNumberResendRequests, [841](#)
  - GevPacketResendMode, [841](#)
  - GevPacketResendTimeout, [841](#)
  - GevResendPacketCount, [841](#)
  - GevResendRequestCount, [842](#)
  - GevTotalPacketCount, [842](#)
  - ICameraBase, [841](#)
  - StreamBlockTransferSize, [842](#)
  - StreamBufferCountManual, [842](#)
  - StreamBufferCountMax, [842](#)
  - StreamBufferCountMode, [842](#)
  - StreamBufferCountResult, [842](#)
  - StreamBufferHandlingMode, [842](#)
  - StreamBufferUnderrunCount, [843](#)
  - StreamCRCCheckEnable, [843](#)
  - StreamDefaultBufferCount, [843](#)
  - StreamDefaultBufferCountMax, [843](#)
  - StreamDefaultBufferCountMode, [843](#)
  - StreamFailedBufferCount, [843](#)
  - StreamID, [843](#)
  - StreamTotalBufferCount, [844](#)
  - StreamType, [844](#)
  - TransportLayerStream, [841](#)
- Spinnaker::Video, [374](#)
- Spinnaker::Video::AVIOption
  - AVIOption, [379](#)
  - frameRate, [379](#)
  - reserved, [379](#)
- Spinnaker::Video::H264Option
  - bitrate, [665](#)
  - frameRate, [665](#)
  - H264Option, [665](#)
  - height, [665](#)
  - reserved, [665](#)
  - width, [665](#)
- Spinnaker::Video::MJPGOption
  - frameRate, [770](#)
  - MJPGOption, [770](#)
  - quality, [770](#)
  - reserved, [770](#)
- Spinnaker::Video::SpinVideo
  - ~SpinVideo, [807](#)
  - Append, [807](#)
  - Close, [807](#)
  - Open, [808](#)
  - SetMaximumFileSize, [809](#)
  - SpinVideo, [807](#)
- SpinnakerLogLevel
  - Spinnaker Definitions, [166](#)
- Standard
  - Types Enums, [313](#)
- StartRecording
  - Spinnaker::GenApi::PortNode, [793](#)
  - Spinnaker::GenApi::PortRecorder, [796](#)
- StatisticsChannel
  - Spinnaker Definitions, [166](#)
- Status
  - Spinnaker::ActionCommandResult, [375](#)
- StopRecording
  - IPortRecorder Interface, [271](#)
  - Spinnaker::GenApi::PortNode, [793](#)
  - Spinnaker::GenApi::PortRecorder, [796](#)
- StoreToBag
  - Spinnaker::GenApi::CFeatureBag, [559](#)
- Stream
  - Spinnaker::Event, [637](#)
  - Spinnaker::Image, [723](#)
- StreamBlockTransferSize
  - Spinnaker::TransportLayerStream, [842](#)
- StreamBufferCountManual
  - Spinnaker::TransportLayerStream, [842](#)
- StreamBufferCountMax
  - Spinnaker::TransportLayerStream, [842](#)
- StreamBufferCountMode
  - Spinnaker::TransportLayerStream, [842](#)
- StreamBufferCountMode\_Auto
  - TransportLayerDefs Class, [178](#)
- StreamBufferCountMode\_Manual
  - TransportLayerDefs Class, [178](#)
- StreamBufferCountModeEnum
  - TransportLayerDefs Class, [177](#)
- StreamBufferCountResult
  - Spinnaker::TransportLayerStream, [842](#)
- StreamBufferHandlingMode
  - Spinnaker::TransportLayerStream, [842](#)
- StreamBufferHandlingMode\_NewestFirst

- TransportLayerDefs Class, [178](#)
- StreamBufferHandlingMode\_NewestFirstOverwrite
  - TransportLayerDefs Class, [178](#)
- StreamBufferHandlingMode\_NewestOnly
  - TransportLayerDefs Class, [178](#)
- StreamBufferHandlingMode\_OldestFirst
  - TransportLayerDefs Class, [178](#)
- StreamBufferHandlingMode\_OldestFirstOverwrite
  - TransportLayerDefs Class, [178](#)
- StreamBufferHandlingModeEnum
  - TransportLayerDefs Class, [178](#)
- StreamBufferUnderrunCount
  - Spinnaker::TransportLayerStream, [843](#)
- StreamCRCCheckEnable
  - Spinnaker::TransportLayerStream, [843](#)
- StreamChannelId
  - GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [660](#)
  - GVCP\_EVENT\_ITEM, [658](#)
- StreamDefaultBufferCount
  - Spinnaker::TransportLayerStream, [843](#)
- StreamDefaultBufferCountMax
  - Spinnaker::TransportLayerStream, [843](#)
- StreamDefaultBufferCountMode
  - Spinnaker::TransportLayerStream, [843](#)
- StreamDefaultBufferCountMode\_Auto
  - TransportLayerDefs Class, [178](#)
- StreamDefaultBufferCountMode\_Manual
  - TransportLayerDefs Class, [178](#)
- StreamDefaultBufferCountModeEnum
  - TransportLayerDefs Class, [178](#)
- StreamFailedBufferCount
  - Spinnaker::TransportLayerStream, [843](#)
- StreamID
  - Spinnaker::TransportLayerStream, [843](#)
- StreamTotalBufferCount
  - Spinnaker::TransportLayerStream, [844](#)
- StreamType
  - Spinnaker::TransportLayerStream, [844](#)
- StreamType\_CLHS
  - TransportLayerDefs Class, [179](#)
- StreamType\_CXP
  - TransportLayerDefs Class, [179](#)
- StreamType\_CL
  - TransportLayerDefs Class, [179](#)
- StreamType\_Custom
  - TransportLayerDefs Class, [179](#)
- StreamType\_ETHERNET
  - TransportLayerDefs Class, [179](#)
- StreamType\_GEV
  - TransportLayerDefs Class, [179](#)
- StreamType\_IIDC
  - TransportLayerDefs Class, [179](#)
- StreamType\_Mixed
  - TransportLayerDefs Class, [179](#)
- StreamType\_PCI
  - TransportLayerDefs Class, [179](#)
- StreamType\_U3V
  - TransportLayerDefs Class, [179](#)
- StreamType\_UVC
  - TransportLayerDefs Class, [179](#)
- StreamTypeEnum
  - TransportLayerDefs Class, [178](#)
- StringList\_t
  - Types Enums, [310](#)
- StringNode, [809](#)
  - Spinnaker::GenApi::StringNode, [811](#)
- StringNode Class, [303](#)
  - CStringRef, [303](#)
- StringRegNode, [813](#)
  - Spinnaker::GenApi::StringRegNode, [814](#)
- StringRegNode Class, [304](#)
- StructPort Class, [305](#)
- SubMinor
  - Spinnaker::GenICam::Version\_t, [850](#)
- substr
  - Spinnaker::GenICam::gcstring, [657](#)
- swap
  - Spinnaker::GenICam::gcstring, [657](#)
- sync
  - Spinnaker::GenApi::ODevFileStreamBuf, [788](#)
- Synch Class, [306](#)
- System, [815](#)
  - Spinnaker::System, [817](#)
- System Class, [171](#)
- System.h
  - FLIR\_SPINNAKER\_VERSION\_BUILD, [1100](#)
  - FLIR\_SPINNAKER\_VERSION\_MAJOR, [1100](#)
  - FLIR\_SPINNAKER\_VERSION\_MINOR, [1100](#)
  - FLIR\_SPINNAKER\_VERSION\_TYPE, [1100](#)
- SystemImpl
  - Spinnaker::InterfaceList, [748](#)
  - Spinnaker::LoggingEventData, [765](#)
- SystemPtr, [822](#)
  - Spinnaker::SystemPtr, [823](#)
- SystemPtr Class, [172](#)
- TAG\_EVENT\_CMD
  - Spinnaker::GenApi, [371](#)
- TAG\_EVENTDATA\_CMD
  - Spinnaker::GenApi, [371](#)
- TIFFOption, [823](#)
  - Spinnaker::TIFFOption, [824](#)
- TIFF
  - Spinnaker Definitions, [163](#)
- TLDevice
  - Spinnaker::ICameraBase, [672](#)
- TLInterface
  - Spinnaker::IInterface, [698](#)
- TLParamsLocked
  - Spinnaker::Camera, [505](#)
- TLStream
  - Spinnaker::ICameraBase, [672](#)
- Test0001
  - Spinnaker::Camera, [503](#)
- TestEventGenerate
  - Spinnaker::Camera, [503](#)
- TestPattern



- Spinnaker::Camera, [503](#)
- TestPattern\_Increment
  - CameraDefs Class, [127](#)
- TestPattern\_Off
  - CameraDefs Class, [127](#)
- TestPattern\_SensorTestPattern
  - CameraDefs Class, [127](#)
- TestPatternEnums
  - CameraDefs Class, [127](#)
- TestPatternGeneratorSelector
  - Spinnaker::Camera, [503](#)
- TestPatternGeneratorSelector\_PipelineStart
  - CameraDefs Class, [127](#)
- TestPatternGeneratorSelector\_Sensor
  - CameraDefs Class, [127](#)
- TestPatternGeneratorSelectorEnums
  - CameraDefs Class, [127](#)
- TestPendingAck
  - Spinnaker::Camera, [503](#)
- ThrowBadAlloc
  - Spinnaker::GenlCam, [374](#)
- TimerDelay
  - Spinnaker::Camera, [503](#)
- TimerDuration
  - Spinnaker::Camera, [504](#)
- TimerReset
  - Spinnaker::Camera, [504](#)
- TimerSelector
  - Spinnaker::Camera, [504](#)
- TimerSelector\_Timer0
  - CameraDefs Class, [127](#)
- TimerSelector\_Timer1
  - CameraDefs Class, [127](#)
- TimerSelector\_Timer2
  - CameraDefs Class, [127](#)
- TimerSelectorEnums
  - CameraDefs Class, [127](#)
- TimerStatus
  - Spinnaker::Camera, [504](#)
- TimerStatus\_TimerActive
  - CameraDefs Class, [128](#)
- TimerStatus\_TimerCompleted
  - CameraDefs Class, [128](#)
- TimerStatus\_TimerIdle
  - CameraDefs Class, [128](#)
- TimerStatus\_TimerTriggerWait
  - CameraDefs Class, [128](#)
- TimerStatusEnums
  - CameraDefs Class, [127](#)
- TimerTriggerActivation
  - Spinnaker::Camera, [504](#)
- TimerTriggerActivation\_AnyEdge
  - CameraDefs Class, [128](#)
- TimerTriggerActivation\_FallingEdge
  - CameraDefs Class, [128](#)
- TimerTriggerActivation\_LevelHigh
  - CameraDefs Class, [128](#)
- TimerTriggerActivation\_LevelLow
  - CameraDefs Class, [128](#)
- TimerTriggerActivation\_RisingEdge
  - CameraDefs Class, [128](#)
- TimerTriggerActivationEnums
  - CameraDefs Class, [128](#)
- TimerTriggerSource
  - Spinnaker::Camera, [504](#)
- TimerTriggerSource\_AcquisitionEnd
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_AcquisitionStart
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_AcquisitionTrigger
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_Action0
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Action1
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Action2
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Counter0End
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Counter0Start
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Counter1End
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Counter1Start
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Counter2End
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Counter2Start
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Encoder0
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Encoder1
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Encoder2
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_ExposureEnd
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_ExposureStart
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_FrameBurstEnd
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_FrameBurstStart
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_FrameEnd
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_FrameStart
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_FrameTrigger
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_Line0
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_Line1
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Line2
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_LineEnd

- CameraDefs Class, [128](#)
- TimerTriggerSource\_LineStart
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_LineTrigger
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_LinkTrigger0
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_LinkTrigger1
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_LinkTrigger2
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Off
  - CameraDefs Class, [128](#)
- TimerTriggerSource\_SoftwareSignal0
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_SoftwareSignal1
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_SoftwareSignal2
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Timer0End
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Timer0Start
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Timer1End
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Timer1Start
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Timer2End
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_Timer2Start
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_UserOutput0
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_UserOutput1
  - CameraDefs Class, [129](#)
- TimerTriggerSource\_UserOutput2
  - CameraDefs Class, [129](#)
- TimerTriggerSourceEnums
  - CameraDefs Class, [128](#)
- TimerValue
  - Spinnaker::Camera, [504](#)
- Timestamp
  - Spinnaker::Camera, [505](#)
  - U3V\_EVENT\_DATA, [846](#)
- TimestampHigh
  - GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [660](#)
  - GVCP\_EVENT\_ITEM, [658](#)
- TimestampLatch
  - Spinnaker::Camera, [505](#)
- TimestampLatchValue
  - Spinnaker::Camera, [505](#)
- TimestampLow
  - GVCP\_EVENT\_ITEM\_EXTENDED\_ID, [660](#)
  - GVCP\_EVENT\_ITEM, [658](#)
- TimestampReset
  - Spinnaker::Camera, [505](#)
- ToString
  - ISelectorDigit Interface, [276](#)
- IValue Class, [278](#)
- Spinnaker::GenApi::CNodeMapFactory, [589](#)
- Spinnaker::GenApi::CSelectorSet, [611](#)
- Spinnaker::GenApi::EAccessModeClass, [619](#)
- Spinnaker::GenApi::ECachingModeClass, [620](#)
- Spinnaker::GenApi::EDisplayNotationClass, [621](#)
- Spinnaker::GenApi::EEndianessClass, [622](#)
- Spinnaker::GenApi::EGenApiSchemaVersion↔  
Class, [623](#)
- Spinnaker::GenApi::EInputDirectionClass, [623](#),  
[624](#)
- Spinnaker::GenApi::ENameSpaceClass, [624](#)
- Spinnaker::GenApi::ERepresentationClass, [632](#)
- Spinnaker::GenApi::ESignClass, [633](#)
- Spinnaker::GenApi::ESlopeClass, [634](#)
- Spinnaker::GenApi::EStandardNameSpaceClass,  
[635](#)
- Spinnaker::GenApi::EVisibilityClass, [638](#)
- Spinnaker::GenApi::EYesNoClass, [642](#)
- Spinnaker::GenApi::ValueNode, [849](#)
- ToXml
  - Spinnaker::GenApi::CNodeMapFactory, [589](#)
- Tokenize
  - GCUtilities Utility, [234](#)
- TransferAbort
  - Spinnaker::Camera, [505](#)
- TransferBlockCount
  - Spinnaker::Camera, [505](#)
- TransferBurstCount
  - Spinnaker::Camera, [506](#)
- TransferComponentSelector
  - Spinnaker::Camera, [506](#)
- TransferComponentSelector\_All
  - CameraDefs Class, [130](#)
- TransferComponentSelector\_Blue
  - CameraDefs Class, [130](#)
- TransferComponentSelector\_Green
  - CameraDefs Class, [130](#)
- TransferComponentSelector\_Red
  - CameraDefs Class, [130](#)
- TransferComponentSelectorEnums
  - CameraDefs Class, [129](#)
- TransferControlMode
  - Spinnaker::Camera, [506](#)
- TransferControlMode\_Automatic
  - CameraDefs Class, [130](#)
- TransferControlMode\_Basic
  - CameraDefs Class, [130](#)
- TransferControlMode\_UserControlled
  - CameraDefs Class, [130](#)
- TransferControlModeEnums
  - CameraDefs Class, [130](#)
- TransferOperationMode
  - Spinnaker::Camera, [506](#)
- TransferOperationMode\_Continuous
  - CameraDefs Class, [130](#)
- TransferOperationMode\_MultiBlock
  - CameraDefs Class, [130](#)



- TransferOperationModeEnums
  - CameraDefs Class, [130](#)
- TransferPause
  - Spinnaker::Camera, [506](#)
- TransferQueueCurrentBlockCount
  - Spinnaker::Camera, [506](#)
- TransferQueueMaxBlockCount
  - Spinnaker::Camera, [506](#)
- TransferQueueMode
  - Spinnaker::Camera, [507](#)
- TransferQueueMode\_FirstInFirstOut
  - CameraDefs Class, [130](#)
- TransferQueueModeEnums
  - CameraDefs Class, [130](#)
- TransferQueueOverflowCount
  - Spinnaker::Camera, [507](#)
- TransferResume
  - Spinnaker::Camera, [507](#)
- TransferSelector
  - Spinnaker::Camera, [507](#)
- TransferSelector\_All
  - CameraDefs Class, [131](#)
- TransferSelector\_Stream0
  - CameraDefs Class, [131](#)
- TransferSelector\_Stream1
  - CameraDefs Class, [131](#)
- TransferSelector\_Stream2
  - CameraDefs Class, [131](#)
- TransferSelectorEnums
  - CameraDefs Class, [130](#)
- TransferStart
  - Spinnaker::Camera, [507](#)
- TransferStatus
  - Spinnaker::Camera, [507](#)
- TransferStatusSelector
  - Spinnaker::Camera, [507](#)
- TransferStatusSelector\_Paused
  - CameraDefs Class, [131](#)
- TransferStatusSelector\_QueueOverflow
  - CameraDefs Class, [131](#)
- TransferStatusSelector\_Stopped
  - CameraDefs Class, [131](#)
- TransferStatusSelector\_Stopping
  - CameraDefs Class, [131](#)
- TransferStatusSelector\_Streaming
  - CameraDefs Class, [131](#)
- TransferStatusSelectorEnums
  - CameraDefs Class, [131](#)
- TransferStop
  - Spinnaker::Camera, [507](#)
- TransferStreamChannel
  - Spinnaker::Camera, [508](#)
- TransferTriggerActivation
  - Spinnaker::Camera, [508](#)
- TransferTriggerActivation\_AnyEdge
  - CameraDefs Class, [131](#)
- TransferTriggerActivation\_FallingEdge
  - CameraDefs Class, [131](#)
- TransferTriggerActivation\_LevelHigh
  - CameraDefs Class, [131](#)
- TransferTriggerActivation\_LevelLow
  - CameraDefs Class, [131](#)
- TransferTriggerActivation\_RisingEdge
  - CameraDefs Class, [131](#)
- TransferTriggerActivationEnums
  - CameraDefs Class, [131](#)
- TransferTriggerMode
  - Spinnaker::Camera, [508](#)
- TransferTriggerMode\_Off
  - CameraDefs Class, [132](#)
- TransferTriggerMode\_On
  - CameraDefs Class, [132](#)
- TransferTriggerModeEnums
  - CameraDefs Class, [131](#)
- TransferTriggerSelector
  - Spinnaker::Camera, [508](#)
- TransferTriggerSelector\_TransferAbort
  - CameraDefs Class, [132](#)
- TransferTriggerSelector\_TransferActive
  - CameraDefs Class, [132](#)
- TransferTriggerSelector\_TransferBurstStart
  - CameraDefs Class, [132](#)
- TransferTriggerSelector\_TransferBurstStop
  - CameraDefs Class, [132](#)
- TransferTriggerSelector\_TransferPause
  - CameraDefs Class, [132](#)
- TransferTriggerSelector\_TransferResume
  - CameraDefs Class, [132](#)
- TransferTriggerSelector\_TransferStart
  - CameraDefs Class, [132](#)
- TransferTriggerSelector\_TransferStop
  - CameraDefs Class, [132](#)
- TransferTriggerSelectorEnums
  - CameraDefs Class, [132](#)
- TransferTriggerSource
  - Spinnaker::Camera, [508](#)
- TransferTriggerSource\_Action0
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Action1
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Action2
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Counter0End
  - CameraDefs Class, [132](#)
- TransferTriggerSource\_Counter0Start
  - CameraDefs Class, [132](#)
- TransferTriggerSource\_Counter1End
  - CameraDefs Class, [132](#)
- TransferTriggerSource\_Counter1Start
  - CameraDefs Class, [132](#)
- TransferTriggerSource\_Counter2End
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Counter2Start
  - CameraDefs Class, [132](#)
- TransferTriggerSource\_Line0
  - CameraDefs Class, [132](#)

- TransferTriggerSource\_Line1
  - CameraDefs Class, [132](#)
- TransferTriggerSource\_Line2
  - CameraDefs Class, [132](#)
- TransferTriggerSource\_SoftwareSignal0
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_SoftwareSignal1
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_SoftwareSignal2
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Timer0End
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Timer0Start
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Timer1End
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Timer1Start
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Timer2End
  - CameraDefs Class, [133](#)
- TransferTriggerSource\_Timer2Start
  - CameraDefs Class, [133](#)
- TransferTriggerSourceEnums
  - CameraDefs Class, [132](#)
- TransportLayerDefs Class, [174](#)
  - DeviceAccessStatus\_NoAccess, [175](#)
  - DeviceAccessStatus\_ReadOnly, [175](#)
  - DeviceAccessStatus\_ReadWrite, [175](#)
  - DeviceAccessStatus\_Unknown, [175](#)
  - DeviceAccessStatusEnum, [175](#)
  - DeviceCurrentSpeed\_FullSpeed, [176](#)
  - DeviceCurrentSpeed\_HighSpeed, [176](#)
  - DeviceCurrentSpeed\_LowSpeed, [176](#)
  - DeviceCurrentSpeed\_SuperSpeed, [176](#)
  - DeviceCurrentSpeed\_UnknownSpeed, [176](#)
  - DeviceCurrentSpeedEnum, [175](#)
  - DeviceEndiannessMechanism\_Legacy, [176](#)
  - DeviceEndiannessMechanism\_Standard, [176](#)
  - DeviceEndiannessMechanismEnum, [176](#)
  - DeviceType\_CLHS, [176](#)
  - DeviceType\_CXP, [176](#)
  - DeviceType\_CL, [176](#)
  - DeviceType\_Custom, [176](#)
  - DeviceType\_ETHERNET, [176](#)
  - DeviceType\_GEV, [176](#)
  - DeviceType\_IIDC, [176](#)
  - DeviceType\_Mixed, [176](#)
  - DeviceType\_PCI, [176](#)
  - DeviceType\_U3V, [176](#)
  - DeviceType\_UVC, [176](#)
  - DeviceTypeEnum, [176](#)
  - GUIXMLLocation\_Device, [177](#)
  - GUIXMLLocation\_Host, [177](#)
  - GUIXMLLocationEnum, [177](#)
  - GenICamXMLLocation\_Device, [177](#)
  - GenICamXMLLocation\_Host, [177](#)
  - GenICamXMLLocationEnum, [176](#)
  - GevCCP\_EnumEntry\_GevCCP\_ControlAccess, [177](#)
  - GevCCP\_EnumEntry\_GevCCP\_ExclusiveAccess, [177](#)
  - GevCCP\_EnumEntry\_GevCCP\_OpenAccess, [177](#)
  - GevCCPEnum, [177](#)
  - NUMDEVICEACCESSSTATUS, [175](#)
  - NUMDEVICECURRENTSPEED, [176](#)
  - NUMDEVICEENDIANESSMECHANISM, [176](#)
  - NUMDEVICETYPE, [176](#)
  - NUMGENICAMXMLLOCATION, [177](#)
  - NUMGEVCCP, [177](#)
  - NUMGUIXMLLOCATION, [177](#)
  - NUMPOESTATUS, [177](#)
  - NUMSTREAMBUFFERCOUNTMODE, [178](#)
  - NUMSTREAMBUFFERHANDLINGMODE, [178](#)
  - NUMSTREAMDEFAULTBUFFERCOUNTMODE, [178](#)
  - NUMSTREAMTYPE, [179](#)
  - POEStatus\_NotSupported, [177](#)
  - POEStatus\_PowerOff, [177](#)
  - POEStatus\_PowerOn, [177](#)
  - POEStatusEnum, [177](#)
  - StreamBufferCountMode\_Auto, [178](#)
  - StreamBufferCountMode\_Manual, [178](#)
  - StreamBufferCountModeEnum, [177](#)
  - StreamBufferHandlingMode\_NewestFirst, [178](#)
  - StreamBufferHandlingMode\_NewestFirstOverwrite, [178](#)
  - StreamBufferHandlingMode\_NewestOnly, [178](#)
  - StreamBufferHandlingMode\_OldestFirst, [178](#)
  - StreamBufferHandlingMode\_OldestFirstOverwrite, [178](#)
  - StreamBufferHandlingModeEnum, [178](#)
  - StreamDefaultBufferCountMode\_Auto, [178](#)
  - StreamDefaultBufferCountMode\_Manual, [178](#)
  - StreamDefaultBufferCountModeEnum, [178](#)
  - StreamType\_CLHS, [179](#)
  - StreamType\_CXP, [179](#)
  - StreamType\_CL, [179](#)
  - StreamType\_Custom, [179](#)
  - StreamType\_ETHERNET, [179](#)
  - StreamType\_GEV, [179](#)
  - StreamType\_IIDC, [179](#)
  - StreamType\_Mixed, [179](#)
  - StreamType\_PCI, [179](#)
  - StreamType\_U3V, [179](#)
  - StreamType\_UVC, [179](#)
  - StreamTypeEnum, [178](#)
  - TransportLayerDevice, [825](#)
    - Spinnaker::TransportLayerDevice, [827](#)
  - TransportLayerDevice Class, [180](#)
  - TransportLayerInterface, [832](#)
    - Spinnaker::TransportLayerInterface, [834](#)
  - TransportLayerInterface Class, [181](#)
  - TransportLayerStream, [839](#)
    - Spinnaker::TransportLayerStream, [841](#)
  - TransportLayerStream Class, [182](#)

- TriggerActivation
  - Spinnaker::Camera, [508](#)
- TriggerActivation\_AnyEdge
  - CameraDefs Class, [133](#)
- TriggerActivation\_FallingEdge
  - CameraDefs Class, [133](#)
- TriggerActivation\_LevelHigh
  - CameraDefs Class, [133](#)
- TriggerActivation\_LevelLow
  - CameraDefs Class, [133](#)
- TriggerActivation\_RisingEdge
  - CameraDefs Class, [133](#)
- TriggerActivationEnums
  - CameraDefs Class, [133](#)
- TriggerDelay
  - Spinnaker::Camera, [508](#)
- TriggerDivider
  - Spinnaker::Camera, [509](#)
- TriggerEventTest
  - Spinnaker::Camera, [509](#)
- TriggerMode
  - Spinnaker::Camera, [509](#)
- TriggerMode\_Off
  - CameraDefs Class, [133](#)
- TriggerMode\_On
  - CameraDefs Class, [133](#)
- TriggerModeEnums
  - CameraDefs Class, [133](#)
- TriggerMultiplier
  - Spinnaker::Camera, [509](#)
- TriggerOverlap
  - Spinnaker::Camera, [509](#)
- TriggerOverlap\_Off
  - CameraDefs Class, [134](#)
- TriggerOverlap\_PreviousFrame
  - CameraDefs Class, [134](#)
- TriggerOverlap\_ReadOut
  - CameraDefs Class, [134](#)
- TriggerOverlapEnums
  - CameraDefs Class, [133](#)
- TriggerSelector
  - Spinnaker::Camera, [509](#)
- TriggerSelector\_AcquisitionStart
  - CameraDefs Class, [134](#)
- TriggerSelector\_FrameBurstStart
  - CameraDefs Class, [134](#)
- TriggerSelector\_FrameStart
  - CameraDefs Class, [134](#)
- TriggerSelectorEnums
  - CameraDefs Class, [134](#)
- TriggerSoftware
  - Spinnaker::Camera, [509](#)
- TriggerSource
  - Spinnaker::Camera, [510](#)
- TriggerSource\_Action0
  - CameraDefs Class, [134](#)
- TriggerSource\_Counter0End
  - CameraDefs Class, [134](#)
- TriggerSource\_Counter0Start
  - CameraDefs Class, [134](#)
- TriggerSource\_Counter1End
  - CameraDefs Class, [134](#)
- TriggerSource\_Counter1Start
  - CameraDefs Class, [134](#)
- TriggerSource\_Line0
  - CameraDefs Class, [134](#)
- TriggerSource\_Line1
  - CameraDefs Class, [134](#)
- TriggerSource\_Line2
  - CameraDefs Class, [134](#)
- TriggerSource\_Line3
  - CameraDefs Class, [134](#)
- TriggerSource\_LogicBlock0
  - CameraDefs Class, [134](#)
- TriggerSource\_LogicBlock1
  - CameraDefs Class, [134](#)
- TriggerSource\_Software
  - CameraDefs Class, [134](#)
- TriggerSource\_UserOutput0
  - CameraDefs Class, [134](#)
- TriggerSource\_UserOutput1
  - CameraDefs Class, [134](#)
- TriggerSource\_UserOutput2
  - CameraDefs Class, [134](#)
- TriggerSource\_UserOutput3
  - CameraDefs Class, [134](#)
- TriggerSourceEnums
  - CameraDefs Class, [134](#)
- TryLock
  - Spinnaker::GenApi::CLock, [576](#)
  - Spinnaker::GenICam::CGlobalLock, [564](#)
  - Spinnaker::GenICam::CLock, [578](#)
- type
  - Spinnaker::LibraryVersion, [758](#)
- Types Enums, [308](#)
  - \_CycleDetectAccesMode, [311](#)
  - \_UndefinedRepresentation, [310](#)
  - \_Undefined, [312](#)
  - \_UndefinedAccesMode, [311](#)
  - \_UndefinedCachingMode, [311](#)
  - \_UndefinedEDisplayNotation, [311](#)
  - \_UndefinedESlope, [314](#)
  - \_UndefinedEXMLValidation, [314](#)
  - \_UndefinedEndian, [311](#)
  - \_UndefinedNameSpace, [313](#)
  - \_UndefinedRepresentation, [313](#)
  - \_UndefinedSign, [313](#)
  - \_UndefinedStandardNameSpace, [314](#)
  - \_UndefinedVisibility, [314](#)
  - \_UndefinedYesNo, [315](#)
  - Automatic, [314](#)
  - Beginner, [314](#)
  - BigEndian, [311](#)
  - Boolean, [313](#)
  - CL, [314](#)
  - ctDependingNodes, [313](#)

- ctlInvalidatingChildren, 313
- ctParentNodes, 313
- ctReadingChildren, 313
- ctTerminalNodes, 313
- ctWritingChildren, 313
- Custom, 313
- Decreasing, 314
- EAccessMode, 311
- ECachingMode, 311
- EDisplayNotation, 311
- EEndianess, 311
- EGenApiSchemaVersion, 311
- ELncMode, 312
- EInputDirection, 312
- EInterfaceType, 312
- ELinkType, 312
- ENamespace, 313
- ERepresentation, 313
- ESign, 313
- ESlope, 313
- EStandardNameSpace, 314
- EVisibility, 314
- EXMLValidation, 314
- EYesNo, 314
- Expert, 314
- fixedIncrement, 312
- fnAutomatic, 311
- fnFixed, 311
- fnScientific, 311
- GEV, 314
- Guru, 314
- HexNumber, 313
- IIDC, 314
- IPV4Address, 313
- idFrom, 312
- idNone, 312
- idTo, 312
- Increasing, 314
- intflBase, 312
- intflBoolean, 312
- intflCategory, 312
- intflCommand, 312
- intflEnumEntry, 312
- intflEnumeration, 312
- intflFloat, 312
- intflInteger, 312
- intflPort, 312
- intflRegister, 312
- intflString, 312
- intflValue, 312
- Invisible, 314
- Linear, 313
- listIncrement, 312
- LittleEndian, 311
- Logarithmic, 313
- MACAddress, 313
- NA, 311
- NI, 311
- No, 315
- NoCache, 311
- noIncrement, 312
- None, 314
- PureNumber, 313
- RO, 311
- RW, 311
- Signed, 313
- Standard, 313
- StringList\_t, 310
- USB, 314
- Unsigned, 313
- v1\_0, 312
- v1\_1, 312
- Varying, 314
- WO, 311
- WriteAround, 311
- WriteThrough, 311
- xvAll, 314
- xvCycles, 314
- xvDefault, 314
- xvLoad, 314
- xvSFNC, 314
- Yes, 315
- Types.h
  - interface, 1086
- U3V\_CHUNK\_TRAILER, 844
  - ChunkID, 844
  - ChunkLength, 844
- U3V\_COMMAND\_HEADER, 845
  - CommandId, 845
  - Flags, 845
  - Length, 845
  - Prefix, 845
  - ReqId, 845
- U3V\_EVENT\_DATA, 845
  - EventId, 846
  - Reserved, 846
  - Timestamp, 846
- U3V\_EVENT\_MESSAGE, 846
  - CommandHeader, 847
  - EventData, 847
- U3V\_EVENT\_PREFIX
  - Spinnaker::GenApi, 372
- UNKNOWN\_PIXELFORMAT
  - CameraDefs Class, 113
- USE\_TEMP\_CACHE\_FILE
  - GCUtilities.h, 1004
- USB
  - Types Enums, 314
- underflow
  - Spinnaker::GenApi::IDevFileStreamBuf, 683
- Unlock
  - Spinnaker::GenApi::CLock, 576
  - Spinnaker::GenICam::CGlobalLock, 564
  - Spinnaker::GenICam::CLock, 578
- UnlockEarly
  - Spinnaker::GenICam::CGlobalLockUnlocker, 565

- UnregisterAllLoggingEvent
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::System, [820](#)
- UnregisterEvent
  - Spinnaker::CameraBase, [521](#)
  - Spinnaker::ICameraBase, [671](#)
  - Spinnaker::IInterface, [698](#)
  - Spinnaker::Interface, [742](#)
- UnregisterInterfaceEvent
  - Spinnaker::ISystem, [755](#)
  - Spinnaker::System, [821](#)
- UnregisterLoggingEvent
  - Spinnaker::ISystem, [756](#)
  - Spinnaker::System, [821](#)
- Unsigned
  - Types Enums, [313](#)
- UpdateBuffer
  - Spinnaker::GenApi::CChunkAdapter, [531](#)
  - Spinnaker::GenApi::CChunkPort, [541](#)
- UpdateCameras
  - Spinnaker::IInterface, [698](#)
  - Spinnaker::ISystem, [756](#)
  - Spinnaker::Interface, [742](#)
  - Spinnaker::System, [821](#)
- UpdateFirmware
  - SpinUpdate.h, [1096](#)
- UpdateFirmwareConsole
  - SpinUpdate.h, [1096](#)
- UpdatorMessageCallback
  - SpinUpdate.h, [1096](#)
- UpdatorProgressCallback
  - SpinUpdate.h, [1096](#)
- UrlDecode
  - GCUtilities Utility, [234](#)
- UrlEncode
  - GCUtilities Utility, [234](#)
- UserOutputSelector
  - Spinnaker::Camera, [510](#)
- UserOutputSelector\_UserOutput0
  - CameraDefs Class, [135](#)
- UserOutputSelector\_UserOutput1
  - CameraDefs Class, [135](#)
- UserOutputSelector\_UserOutput2
  - CameraDefs Class, [135](#)
- UserOutputSelector\_UserOutput3
  - CameraDefs Class, [135](#)
- UserOutputSelectorEnums
  - CameraDefs Class, [134](#)
- UserOutputValue
  - Spinnaker::Camera, [510](#)
- UserOutputValueAll
  - Spinnaker::Camera, [510](#)
- UserOutputValueAllMask
  - Spinnaker::Camera, [510](#)
- UserSetDefault
  - Spinnaker::Camera, [510](#)
- UserSetDefault\_Default
  - CameraDefs Class, [135](#)
- UserSetDefault\_UserSet0
  - CameraDefs Class, [135](#)
- UserSetDefault\_UserSet1
  - CameraDefs Class, [135](#)
- UserSetDefaultEnums
  - CameraDefs Class, [135](#)
- UserSetFeatureEnable
  - Spinnaker::Camera, [511](#)
- UserSetLoad
  - Spinnaker::Camera, [511](#)
- UserSetSave
  - Spinnaker::Camera, [511](#)
- UserSetSelector
  - Spinnaker::Camera, [511](#)
- UserSetSelector\_Default
  - CameraDefs Class, [135](#)
- UserSetSelector\_UserSet0
  - CameraDefs Class, [135](#)
- UserSetSelector\_UserSet1
  - CameraDefs Class, [135](#)
- UserSetSelectorEnums
  - CameraDefs Class, [135](#)
- v1\_0
  - Types Enums, [312](#)
- v1\_1
  - Types Enums, [312](#)
- V3\_3Enable
  - Spinnaker::Camera, [511](#)
- ValueNode, [847](#)
  - Spinnaker::GenApi::ValueNode, [848](#)
- ValueNode Class, [316](#)
  - CValueRef, [316](#)
- Varying
  - Types Enums, [314](#)
- Verify
  - IBoolean Interface, [236](#)
- Version\_t, [849](#)
- WEIGHTED\_DIRECTIONAL\_FILTER
  - Spinnaker Definitions, [161](#)
- what
  - Spinnaker::Exception, [641](#)
- WhiteClip
  - Spinnaker::Camera, [511](#)
- WhiteClipSelector
  - Spinnaker::Camera, [512](#)
- WhiteClipSelector\_All
  - CameraDefs Class, [135](#)
- WhiteClipSelector\_Blue
  - CameraDefs Class, [135](#)
- WhiteClipSelector\_Green
  - CameraDefs Class, [135](#)
- WhiteClipSelector\_Red
  - CameraDefs Class, [135](#)
- WhiteClipSelector\_Tap1
  - CameraDefs Class, [135](#)
- WhiteClipSelector\_Tap2
  - CameraDefs Class, [135](#)

- WhiteClipSelector\_U
  - CameraDefs Class, [135](#)
- WhiteClipSelector\_V
  - CameraDefs Class, [135](#)
- WhiteClipSelector\_Y
  - CameraDefs Class, [135](#)
- WhiteClipSelectorEnums
  - CameraDefs Class, [135](#)
- Width
  - Spinnaker::Camera, [512](#)
- width
  - Spinnaker::Video::H264Option, [665](#)
- WidthMax
  - Spinnaker::Camera, [512](#)
- WO
  - Types Enums, [311](#)
- Write
  - IPort Interface, [268](#)
  - Spinnaker::GenApi::CChunkPort, [541](#)
  - Spinnaker::GenApi::CEventPort, [557](#)
  - Spinnaker::GenApi::CPortImpl, [604](#)
  - Spinnaker::GenApi::CPortWriteList, [606](#)
  - Spinnaker::GenApi::CRegisterPortImpl, [609](#)
  - Spinnaker::GenApi::CTestPortStruct, [614](#)
  - Spinnaker::GenApi::PortNode, [793](#)
- write
  - Spinnaker::GenApi::FileProtocolAdapter, [644](#)
- WriteAround
  - Types Enums, [311](#)
- WritePort
  - Spinnaker::CameraBase, [521](#)
  - Spinnaker::ICameraBase, [671](#)
- WriteRegister
  - Spinnaker::GenApi::CRegisterPortImpl, [609](#)
- WriteThrough
  - Types Enums, [311](#)
- xspuIn
  - Spinnaker::GenApi::ODevFileStreamBuf, [788](#)
- xvAll
  - Types Enums, [314](#)
- xvCycles
  - Types Enums, [314](#)
- xvDefault
  - Types Enums, [314](#)
- xvLoad
  - Types Enums, [314](#)
- xvSFNC
  - Types Enums, [314](#)
- Yes
  - Types Enums, [315](#)