

Spinnaker C

3.0.0.118

Generated by Doxygen 1.8.17

1 Getting Started	1
2 Programmer's Guide	3
3 Benefits of Spinnaker	5
4 FlyCapture2 Feature Comparison with Spinnaker	7
5 Working with GenICam GenTL Devices	9
5.1 GenTL Overview	9
5.2 Installation	9
5.3 Troubleshooting	10
5.3.1 Enable FLIR GenTL Logging	10
5.3.2 USB3 Device Image Tearing	10
6 Software Licensing Information	11
7 Software Maintenance Policy	13
7.1 GenTL Overview	13
7.2 Platform Support Policy	13
7.2.1 Windows Support	13
7.2.2 Linux Desktop Support	13
7.2.3 Linux Embedded Support	13
7.2.4 MacOS Support	14
7.3 Versioning Policy	14
8 Module Index	15
8.1 Modules	15
9 Data Structure Index	17
9.1 Data Structures	17
10 File Index	19
10.1 File List	19
11 Module Documentation	21
11.1 Spinnaker C Definitions	21
11.2 Camera Enumerations	22
11.3 Chunk Data Structures	23
11.4 Spinnaker C QuickSpin API	24
11.4.1 Detailed Description	24
11.5 QuickSpin Access	25
11.6 Spinnaker C API	26
11.6.1 Detailed Description	26
11.7 Error Handling	27
11.8 System Access	28

11.9 InterfaceList Access	29
11.10 CameraList Access	30
11.11 ImageList Access	31
11.12 Interface Access	32
11.13 Camera Access	33
11.14 Image Access	34
11.15 Image Processor Access	35
11.16 Event Access	38
11.17 ImageStatistics Access	39
11.18 Logging Event Data Access	40
11.19 Device Event Data Access	41
11.20 Chunk data access	42
11.21 Spinnaker C Handles	43
11.22 Spinnaker C Function Signatures	44
11.23 Spinnaker C Enumerations	45
11.24 Spinnaker C Structures	46
11.25 Spinnaker C GenICam API	47
11.26 Node Map Access	48
11.27 Node Access	49
11.28 IValue Access	50
11.29 String Access	51
11.30 IInteger Access	52
11.31 IFloat Access	53
11.32 IEnumeration Access	54
11.33 IEnumEntry Access	55
11.34 IBoolean Access	56
11.35 ICommand Access	57
11.36 ICategory Access	58
11.37 IRegister Access	59
11.38 Spinnaker C GenICam Handles	60
11.39 Spinnaker C GenICam Enumerations	61
11.40 SpinVideo Recording Access	62
11.41 Transport Layer Enumerations	63
11.42 TLDevice Structures	64
11.43 TLInterface Structures	65
11.44 TLStream Structures	66
11.45 TLSystem Structures	67
12 Data Structure Documentation	69
12.1 _actionCommandResult Struct Reference	69
12.1.1 Detailed Description	69
12.1.2 Field Documentation	69

12.1.2.1 DeviceAddress	69
12.1.2.2 Status	69
12.2 _quickSpin Struct Reference	70
12.2.1 Field Documentation	82
12.2.1.1 AasRoiEnable	82
12.2.1.2 AasRoiHeight	82
12.2.1.3 AasRoiOffsetX	82
12.2.1.4 AasRoiOffsetY	82
12.2.1.5 AasRoiWidth	83
12.2.1.6 AcquisitionAbort	83
12.2.1.7 AcquisitionArm	83
12.2.1.8 AcquisitionBurstFrameCount	83
12.2.1.9 AcquisitionFrameCount	83
12.2.1.10 AcquisitionFrameRate	83
12.2.1.11 AcquisitionFrameRateEnable	83
12.2.1.12 AcquisitionLineRate	83
12.2.1.13 AcquisitionMode	84
12.2.1.14 AcquisitionResultingFrameRate	84
12.2.1.15 AcquisitionStart	84
12.2.1.16 AcquisitionStatus	84
12.2.1.17 AcquisitionStatusSelector	84
12.2.1.18 AcquisitionStop	84
12.2.1.19 ActionDeviceKey	84
12.2.1.20 ActionGroupKey	84
12.2.1.21 ActionGroupMask	85
12.2.1.22 ActionQueueSize	85
12.2.1.23 ActionSelector	85
12.2.1.24 ActionUnconditionalMode	85
12.2.1.25 AdaptiveCompressionEnable	85
12.2.1.26 AdcBitDepth	85
12.2.1.27 aPAUSEMACCtrlFramesReceived	85
12.2.1.28 aPAUSEMACCtrlFramesTransmitted	85
12.2.1.29 AutoAlgorithmSelector	86
12.2.1.30 AutoExposureControlLoopDamping	86
12.2.1.31 AutoExposureControlPriority	86
12.2.1.32 AutoExposureEVCompensation	86
12.2.1.33 AutoExposureExposureTimeLowerLimit	86
12.2.1.34 AutoExposureExposureTimeUpperLimit	86
12.2.1.35 AutoExposureGainLowerLimit	86
12.2.1.36 AutoExposureGainUpperLimit	86
12.2.1.37 AutoExposureGreyValueLowerLimit	87
12.2.1.38 AutoExposureGreyValueUpperLimit	87

12.2.1.39 AutoExposureLightingMode	87
12.2.1.40 AutoExposureMeteringMode	87
12.2.1.41 AutoExposureTargetGreyValue	87
12.2.1.42 AutoExposureTargetGreyValueAuto	87
12.2.1.43 BalanceRatio	87
12.2.1.44 BalanceRatioSelector	87
12.2.1.45 BalanceWhiteAuto	88
12.2.1.46 BalanceWhiteAutoDamping	88
12.2.1.47 BalanceWhiteAutoLowerLimit	88
12.2.1.48 BalanceWhiteAutoProfile	88
12.2.1.49 BalanceWhiteAutoUpperLimit	88
12.2.1.50 BinningHorizontal	88
12.2.1.51 BinningHorizontalMode	88
12.2.1.52 BinningSelector	88
12.2.1.53 BinningVertical	89
12.2.1.54 BinningVerticalMode	89
12.2.1.55 BlackLevel	89
12.2.1.56 BlackLevelAuto	89
12.2.1.57 BlackLevelAutoBalance	89
12.2.1.58 BlackLevelClampingEnable	89
12.2.1.59 BlackLevelRaw	89
12.2.1.60 BlackLevelSelector	89
12.2.1.61 ChunkBlackLevel	90
12.2.1.62 ChunkBlackLevelSelector	90
12.2.1.63 ChunkCompressionMode	90
12.2.1.64 ChunkCompressionRatio	90
12.2.1.65 ChunkCounterSelector	90
12.2.1.66 ChunkCounterValue	90
12.2.1.67 ChunkCRC	90
12.2.1.68 ChunkEnable	90
12.2.1.69 ChunkEncoderSelector	91
12.2.1.70 ChunkEncoderStatus	91
12.2.1.71 ChunkEncoderValue	91
12.2.1.72 ChunkExposureEndLineStatusAll	91
12.2.1.73 ChunkExposureTime	91
12.2.1.74 ChunkExposureTimeSelector	91
12.2.1.75 ChunkFrameID	91
12.2.1.76 ChunkGain	91
12.2.1.77 ChunkGainSelector	92
12.2.1.78 ChunkHeight	92
12.2.1.79 ChunkImage	92
12.2.1.80 ChunkImageComponent	92

12.2.1.81 ChunkInferenceBoundingBoxResult	92
12.2.1.82 ChunkInferenceConfidence	92
12.2.1.83 ChunkInferenceFrameId	92
12.2.1.84 ChunkInferenceResult	92
12.2.1.85 ChunkLinePitch	93
12.2.1.86 ChunkLineStatusAll	93
12.2.1.87 ChunkModeActive	93
12.2.1.88 ChunkOffsetX	93
12.2.1.89 ChunkOffsetY	93
12.2.1.90 ChunkPartSelector	93
12.2.1.91 ChunkPixelDynamicRangeMax	93
12.2.1.92 ChunkPixelDynamicRangeMin	93
12.2.1.93 ChunkPixelFormat	94
12.2.1.94 ChunkRegionID	94
12.2.1.95 ChunkScan3dAxisMax	94
12.2.1.96 ChunkScan3dAxisMin	94
12.2.1.97 ChunkScan3dCoordinateOffset	94
12.2.1.98 ChunkScan3dCoordinateReferenceSelector	94
12.2.1.99 ChunkScan3dCoordinateReferenceValue	94
12.2.1.100 ChunkScan3dCoordinateScale	94
12.2.1.101 ChunkScan3dCoordinateSelector	95
12.2.1.102 ChunkScan3dCoordinateSystem	95
12.2.1.103 ChunkScan3dCoordinateSystemReference	95
12.2.1.104 ChunkScan3dCoordinateTransformSelector	95
12.2.1.105 ChunkScan3dDistanceUnit	95
12.2.1.106 ChunkScan3dInvalidDataFlag	95
12.2.1.107 ChunkScan3dInvalidDataValue	95
12.2.1.108 ChunkScan3dOutputMode	95
12.2.1.109 ChunkScan3dTransformValue	96
12.2.1.110 ChunkScanLineSelector	96
12.2.1.111 ChunkSelector	96
12.2.1.112 ChunkSequencerSetActive	96
12.2.1.113 ChunkSerialData	96
12.2.1.114 ChunkSerialDataLength	96
12.2.1.115 ChunkSerialReceiveOverflow	96
12.2.1.116 ChunkSourceID	96
12.2.1.117 ChunkStreamChannelID	97
12.2.1.118 ChunkTimerSelector	97
12.2.1.119 ChunkTimerValue	97
12.2.1.120 ChunkTimestamp	97
12.2.1.121 ChunkTimestampLatchValue	97
12.2.1.122 ChunkTransferBlockID	97

12.2.1.123 ChunkTransferQueueCurrentBlockCount	97
12.2.1.124 ChunkTransferStreamID	97
12.2.1.125 ChunkWidth	98
12.2.1.126 CIConfiguration	98
12.2.1.127 CITimeSlotsCount	98
12.2.1.128 ColorTransformationEnable	98
12.2.1.129 ColorTransformationSelector	98
12.2.1.130 ColorTransformationValue	98
12.2.1.131 ColorTransformationValueSelector	98
12.2.1.132 CompressionRatio	98
12.2.1.133 CompressionSaturationPriority	99
12.2.1.134 CounterDelay	99
12.2.1.135 CounterDuration	99
12.2.1.136 CounterEventActivation	99
12.2.1.137 CounterEventSource	99
12.2.1.138 CounterReset	99
12.2.1.139 CounterResetActivation	99
12.2.1.140 CounterResetSource	99
12.2.1.141 CounterSelector	100
12.2.1.142 CounterStatus	100
12.2.1.143 CounterTriggerActivation	100
12.2.1.144 CounterTriggerSource	100
12.2.1.145 CounterValue	100
12.2.1.146 CounterValueAtReset	100
12.2.1.147 CxpConnectionSelector	100
12.2.1.148 CxpConnectionTestErrorCount	100
12.2.1.149 CxpConnectionTestMode	101
12.2.1.150 CxpConnectionTestPacketCount	101
12.2.1.151 CxpLinkConfiguration	101
12.2.1.152 CxpLinkConfigurationPreferred	101
12.2.1.153 CxpLinkConfigurationStatus	101
12.2.1.154 CxpPoCxpAuto	101
12.2.1.155 CxpPoCxpStatus	101
12.2.1.156 CxpPoCxpTripReset	101
12.2.1.157 CxpPoCxpTurnOff	102
12.2.1.158 DecimationHorizontal	102
12.2.1.159 DecimationHorizontalMode	102
12.2.1.160 DecimationSelector	102
12.2.1.161 DecimationVertical	102
12.2.1.162 DecimationVerticalMode	102
12.2.1.163 DefectCorrectionMode	102
12.2.1.164 DefectCorrectStaticEnable	102

12.2.1.165 DefectTableApply	103
12.2.1.166 DefectTableCoordinateX	103
12.2.1.167 DefectTableCoordinateY	103
12.2.1.168 DefectTableFactoryRestore	103
12.2.1.169 DefectTableIndex	103
12.2.1.170 DefectTablePixelCount	103
12.2.1.171 DefectTableSave	103
12.2.1.172 Deinterlacing	103
12.2.1.173 DeviceCharacterSet	104
12.2.1.174 DeviceClockFrequency	104
12.2.1.175 DeviceClockSelector	104
12.2.1.176 DeviceConnectionSelector	104
12.2.1.177 DeviceConnectionSpeed	104
12.2.1.178 DeviceConnectionStatus	104
12.2.1.179 DeviceEventChannelCount	104
12.2.1.180 DeviceFamilyName	104
12.2.1.181 DeviceFeaturePersistenceEnd	105
12.2.1.182 DeviceFeaturePersistenceStart	105
12.2.1.183 DeviceFirmwareVersion	105
12.2.1.184 DeviceGenCPVersionMajor	105
12.2.1.185 DeviceGenCPVersionMinor	105
12.2.1.186 DeviceID	105
12.2.1.187 DeviceIndicatorMode	105
12.2.1.188 DeviceLinkBandwidthReserve	105
12.2.1.189 DeviceLinkCommandTimeout	106
12.2.1.190 DeviceLinkConnectionCount	106
12.2.1.191 DeviceLinkCurrentThroughput	106
12.2.1.192 DeviceLinkHeartbeatMode	106
12.2.1.193 DeviceLinkHeartbeatTimeout	106
12.2.1.194 DeviceLinkSelector	106
12.2.1.195 DeviceLinkSpeed	106
12.2.1.196 DeviceLinkThroughputLimit	106
12.2.1.197 DeviceLinkThroughputLimitMode	107
12.2.1.198 DeviceManifestEntrySelector	107
12.2.1.199 DeviceManifestPrimaryURL	107
12.2.1.200 DeviceManifestSchemaMajorVersion	107
12.2.1.201 DeviceManifestSchemaMinorVersion	107
12.2.1.202 DeviceManifestSecondaryURL	107
12.2.1.203 DeviceManifestXMLMajorVersion	107
12.2.1.204 DeviceManifestXMLMinorVersion	107
12.2.1.205 DeviceManifestXMLSubMinorVersion	108
12.2.1.206 DeviceManufacturerInfo	108

12.2.1.207 DeviceMaxThroughput	108
12.2.1.208 DeviceModelName	108
12.2.1.209 DevicePowerSupplySelector	108
12.2.1.210 DeviceRegistersCheck	108
12.2.1.211 DeviceRegistersEndianness	108
12.2.1.212 DeviceRegistersStreamingEnd	108
12.2.1.213 DeviceRegistersStreamingStart	109
12.2.1.214 DeviceRegistersValid	109
12.2.1.215 DeviceReset	109
12.2.1.216 DeviceScanType	109
12.2.1.217 DeviceSerialNumber	109
12.2.1.218 DeviceSerialPortBaudRate	109
12.2.1.219 DeviceSerialPortSelector	109
12.2.1.220 DeviceSFNCVersionMajor	109
12.2.1.221 DeviceSFNCVersionMinor	110
12.2.1.222 DeviceSFNCVersionSubMinor	110
12.2.1.223 DeviceStreamChannelCount	110
12.2.1.224 DeviceStreamChannelEndianness	110
12.2.1.225 DeviceStreamChannelLink	110
12.2.1.226 DeviceStreamChannelPacketSize	110
12.2.1.227 DeviceStreamChannelSelector	110
12.2.1.228 DeviceStreamChannelType	110
12.2.1.229 DeviceTapGeometry	111
12.2.1.230 DeviceTemperature	111
12.2.1.231 DeviceTemperatureSelector	111
12.2.1.232 DeviceTLType	111
12.2.1.233 DeviceTLVersionMajor	111
12.2.1.234 DeviceTLVersionMinor	111
12.2.1.235 DeviceTLVersionSubMinor	111
12.2.1.236 DeviceType	111
12.2.1.237 DeviceUptime	112
12.2.1.238 DeviceUserID	112
12.2.1.239 DeviceVendorName	112
12.2.1.240 DeviceVersion	112
12.2.1.241 EncoderDivider	112
12.2.1.242 EncoderMode	112
12.2.1.243 EncoderOutputMode	112
12.2.1.244 EncoderReset	112
12.2.1.245 EncoderResetActivation	113
12.2.1.246 EncoderResetSource	113
12.2.1.247 EncoderSelector	113
12.2.1.248 EncoderSourceA	113

12.2.1.249 EncoderSourceB	113
12.2.1.250 EncoderStatus	113
12.2.1.251 EncoderTimeout	113
12.2.1.252 EncoderValue	113
12.2.1.253 EncoderValueAtReset	114
12.2.1.254 EnumerationCount	114
12.2.1.255 EventAcquisitionEnd	114
12.2.1.256 EventAcquisitionEndFrameID	114
12.2.1.257 EventAcquisitionEndTimestamp	114
12.2.1.258 EventAcquisitionError	114
12.2.1.259 EventAcquisitionErrorFrameID	114
12.2.1.260 EventAcquisitionErrorTimestamp	114
12.2.1.261 EventAcquisitionStart	115
12.2.1.262 EventAcquisitionStartFrameID	115
12.2.1.263 EventAcquisitionStartTimestamp	115
12.2.1.264 EventAcquisitionTransferEnd	115
12.2.1.265 EventAcquisitionTransferEndFrameID	115
12.2.1.266 EventAcquisitionTransferEndTimestamp	115
12.2.1.267 EventAcquisitionTransferStart	115
12.2.1.268 EventAcquisitionTransferStartFrameID	115
12.2.1.269 EventAcquisitionTransferStartTimestamp	116
12.2.1.270 EventAcquisitionTrigger	116
12.2.1.271 EventAcquisitionTriggerFrameID	116
12.2.1.272 EventAcquisitionTriggerTimestamp	116
12.2.1.273 EventActionLate	116
12.2.1.274 EventActionLateFrameID	116
12.2.1.275 EventActionLateTimestamp	116
12.2.1.276 EventCounter0End	116
12.2.1.277 EventCounter0EndFrameID	117
12.2.1.278 EventCounter0EndTimestamp	117
12.2.1.279 EventCounter0Start	117
12.2.1.280 EventCounter0StartFrameID	117
12.2.1.281 EventCounter0StartTimestamp	117
12.2.1.282 EventCounter1End	117
12.2.1.283 EventCounter1EndFrameID	117
12.2.1.284 EventCounter1EndTimestamp	117
12.2.1.285 EventCounter1Start	118
12.2.1.286 EventCounter1StartFrameID	118
12.2.1.287 EventCounter1StartTimestamp	118
12.2.1.288 EventEncoder0Restarted	118
12.2.1.289 EventEncoder0RestartedFrameID	118
12.2.1.290 EventEncoder0RestartedTimestamp	118

12.2.1.291 EventEncoder0Stopped	118
12.2.1.292 EventEncoder0StoppedFrameID	118
12.2.1.293 EventEncoder0StoppedTimestamp	119
12.2.1.294 EventEncoder1 Restarted	119
12.2.1.295 EventEncoder1 RestartedFrameID	119
12.2.1.296 EventEncoder1 RestartedTimestamp	119
12.2.1.297 EventEncoder1 Stopped	119
12.2.1.298 EventEncoder1 StoppedFrameID	119
12.2.1.299 EventEncoder1 StoppedTimestamp	119
12.2.1.300 EventError	119
12.2.1.301 EventErrorCode	120
12.2.1.302 EventErrorFrameID	120
12.2.1.303 EventErrorTimestamp	120
12.2.1.304 EventExposureEnd	120
12.2.1.305 EventExposureEndFrameID	120
12.2.1.306 EventExposureEndTimestamp	120
12.2.1.307 EventExposureStart	120
12.2.1.308 EventExposureStartFrameID	120
12.2.1.309 EventExposureStartTimestamp	121
12.2.1.310 EventFrameBurstEnd	121
12.2.1.311 EventFrameBurstEndFrameID	121
12.2.1.312 EventFrameBurstEndTimestamp	121
12.2.1.313 EventFrameBurstStart	121
12.2.1.314 EventFrameBurstStartFrameID	121
12.2.1.315 EventFrameBurstStartTimestamp	121
12.2.1.316 EventFrameEnd	121
12.2.1.317 EventFrameEndFrameID	122
12.2.1.318 EventFrameEndTimestamp	122
12.2.1.319 EventFrameStart	122
12.2.1.320 EventFrameStartFrameID	122
12.2.1.321 EventFrameStartTimestamp	122
12.2.1.322 EventFrameTransferEnd	122
12.2.1.323 EventFrameTransferEndFrameID	122
12.2.1.324 EventFrameTransferEndTimestamp	122
12.2.1.325 EventFrameTransferStart	123
12.2.1.326 EventFrameTransferStartFrameID	123
12.2.1.327 EventFrameTransferStartTimestamp	123
12.2.1.328 EventFrameTrigger	123
12.2.1.329 EventFrameTriggerFrameID	123
12.2.1.330 EventFrameTriggerTimestamp	123
12.2.1.331 EventLine0AnyEdge	123
12.2.1.332 EventLine0AnyEdgeFrameID	123

12.2.1.333 EventLine0AnyEdgeTimestamp	124
12.2.1.334 EventLine0FallingEdge	124
12.2.1.335 EventLine0FallingEdgeFrameID	124
12.2.1.336 EventLine0FallingEdgeTimestamp	124
12.2.1.337 EventLine0RisingEdge	124
12.2.1.338 EventLine0RisingEdgeFrameID	124
12.2.1.339 EventLine0RisingEdgeTimestamp	124
12.2.1.340 EventLine1AnyEdge	124
12.2.1.341 EventLine1AnyEdgeFrameID	125
12.2.1.342 EventLine1AnyEdgeTimestamp	125
12.2.1.343 EventLine1FallingEdge	125
12.2.1.344 EventLine1FallingEdgeFrameID	125
12.2.1.345 EventLine1FallingEdgeTimestamp	125
12.2.1.346 EventLine1RisingEdge	125
12.2.1.347 EventLine1RisingEdgeFrameID	125
12.2.1.348 EventLine1RisingEdgeTimestamp	125
12.2.1.349 EventLinkSpeedChange	126
12.2.1.350 EventLinkSpeedChangeFrameID	126
12.2.1.351 EventLinkSpeedChangeTimestamp	126
12.2.1.352 EventLinkTrigger0	126
12.2.1.353 EventLinkTrigger0FrameID	126
12.2.1.354 EventLinkTrigger0Timestamp	126
12.2.1.355 EventLinkTrigger1	126
12.2.1.356 EventLinkTrigger1FrameID	126
12.2.1.357 EventLinkTrigger1Timestamp	127
12.2.1.358 EventNotification	127
12.2.1.359 EventSelector	127
12.2.1.360 EventSequencerSetChange	127
12.2.1.361 EventSequencerSetChangeFrameID	127
12.2.1.362 EventSequencerSetChangeTimestamp	127
12.2.1.363 EventSerialData	127
12.2.1.364 EventSerialDataLength	127
12.2.1.365 EventSerialPortReceive	128
12.2.1.366 EventSerialPortReceiveTimestamp	128
12.2.1.367 EventSerialReceiveOverflow	128
12.2.1.368 EventStream0TransferBlockEnd	128
12.2.1.369 EventStream0TransferBlockEndFrameID	128
12.2.1.370 EventStream0TransferBlockEndTimestamp	128
12.2.1.371 EventStream0TransferBlockStart	128
12.2.1.372 EventStream0TransferBlockStartFrameID	128
12.2.1.373 EventStream0TransferBlockStartTimestamp	129
12.2.1.374 EventStream0TransferBlockTrigger	129

12.2.1.375 EventStream0TransferBlockTriggerFrameID	129
12.2.1.376 EventStream0TransferBlockTriggerTimestamp	129
12.2.1.377 EventStream0TransferBurstEnd	129
12.2.1.378 EventStream0TransferBurstEndFrameID	129
12.2.1.379 EventStream0TransferBurstEndTimestamp	129
12.2.1.380 EventStream0TransferBurstStart	129
12.2.1.381 EventStream0TransferBurstStartFrameID	130
12.2.1.382 EventStream0TransferBurstStartTimestamp	130
12.2.1.383 EventStream0TransferEnd	130
12.2.1.384 EventStream0TransferEndFrameID	130
12.2.1.385 EventStream0TransferEndTimestamp	130
12.2.1.386 EventStream0TransferOverflow	130
12.2.1.387 EventStream0TransferOverflowFrameID	130
12.2.1.388 EventStream0TransferOverflowTimestamp	130
12.2.1.389 EventStream0TransferPause	131
12.2.1.390 EventStream0TransferPauseFrameID	131
12.2.1.391 EventStream0TransferPauseTimestamp	131
12.2.1.392 EventStream0TransferResume	131
12.2.1.393 EventStream0TransferResumeFrameID	131
12.2.1.394 EventStream0TransferResumeTimestamp	131
12.2.1.395 EventStream0TransferStart	131
12.2.1.396 EventStream0TransferStartFrameID	131
12.2.1.397 EventStream0TransferStartTimestamp	132
12.2.1.398 EventTest	132
12.2.1.399 EventTestTimestamp	132
12.2.1.400 EventTimer0End	132
12.2.1.401 EventTimer0EndFrameID	132
12.2.1.402 EventTimer0EndTimestamp	132
12.2.1.403 EventTimer0Start	132
12.2.1.404 EventTimer0StartFrameID	132
12.2.1.405 EventTimer0StartTimestamp	133
12.2.1.406 EventTimer1End	133
12.2.1.407 EventTimer1EndFrameID	133
12.2.1.408 EventTimer1EndTimestamp	133
12.2.1.409 EventTimer1Start	133
12.2.1.410 EventTimer1StartFrameID	133
12.2.1.411 EventTimer1StartTimestamp	133
12.2.1.412 ExposureActiveMode	133
12.2.1.413 ExposureAuto	134
12.2.1.414 ExposureMode	134
12.2.1.415 ExposureTime	134
12.2.1.416 ExposureTimeMode	134

12.2.1.417 ExposureTimeSelector	134
12.2.1.418 FactoryReset	134
12.2.1.419 FileAccessBuffer	134
12.2.1.420 FileAccessLength	134
12.2.1.421 FileAccessOffset	135
12.2.1.422 FileOpenMode	135
12.2.1.423 FileOperationExecute	135
12.2.1.424 FileOperationResult	135
12.2.1.425 FileOperationSelector	135
12.2.1.426 FileOperationStatus	135
12.2.1.427 FileSelector	135
12.2.1.428 FileSize	135
12.2.1.429 Gain	136
12.2.1.430 GainAuto	136
12.2.1.431 GainAutoBalance	136
12.2.1.432 GainSelector	136
12.2.1.433 Gamma	136
12.2.1.434 GammaEnable	136
12.2.1.435 GevActiveLinkCount	136
12.2.1.436 GevCCP	136
12.2.1.437 GevCurrentDefaultGateway	137
12.2.1.438 GevCurrentIPAddress	137
12.2.1.439 GevCurrentIPConfigurationDHCP	137
12.2.1.440 GevCurrentIPConfigurationLLA	137
12.2.1.441 GevCurrentIPConfigurationPersistentIP	137
12.2.1.442 GevCurrentPhysicalLinkConfiguration	137
12.2.1.443 GevCurrentSubnetMask	137
12.2.1.444 GevDiscoveryAckDelay	137
12.2.1.445 GevFirstURL	138
12.2.1.446 GevGVCPExtendedStatusCodes	138
12.2.1.447 GevGVCPExtendedStatusCodesSelector	138
12.2.1.448 GevGVCPHeartbeatDisable	138
12.2.1.449 GevGVCPPendingAck	138
12.2.1.450 GevGVCPPendingTimeout	138
12.2.1.451 GevGVSPExtendedIDMode	138
12.2.1.452 GevHeartbeatTimeout	138
12.2.1.453 GevIEEE1588	139
12.2.1.454 GevIEEE1588ClockAccuracy	139
12.2.1.455 GevIEEE1588Mode	139
12.2.1.456 GevIEEE1588Status	139
12.2.1.457 GevInterfaceSelector	139
12.2.1.458 GevIPConfigurationStatus	139

12.2.1.459	GevMACAddress	139
12.2.1.460	GevMCDA	139
12.2.1.461	GevMCPHostPort	140
12.2.1.462	GevMCRC	140
12.2.1.463	GevMCSP	140
12.2.1.464	GevMCTT	140
12.2.1.465	GevNumberOfInterfaces	140
12.2.1.466	GevPAUSEFrameReception	140
12.2.1.467	GevPAUSEFrameTransmission	140
12.2.1.468	GevPersistentDefaultGateway	140
12.2.1.469	GevPersistentIPAddress	141
12.2.1.470	GevPersistentSubnetMask	141
12.2.1.471	GevPhysicalLinkConfiguration	141
12.2.1.472	GevPrimaryApplicationIPAddress	141
12.2.1.473	GevPrimaryApplicationSocket	141
12.2.1.474	GevPrimaryApplicationSwitchoverKey	141
12.2.1.475	GevSCCFGAllInTransmission	141
12.2.1.476	GevSCCFGExtendedChunkData	141
12.2.1.477	GevSCCFGPacketResendDestination	142
12.2.1.478	GevSCCFGUnconditionalStreaming	142
12.2.1.479	GevSCDA	142
12.2.1.480	GevSCPD	142
12.2.1.481	GevSCPDDirection	142
12.2.1.482	GevSCPHostPort	142
12.2.1.483	GevSCPIInterfaceIndex	142
12.2.1.484	GevSCPSBigEndian	142
12.2.1.485	GevSCPSDoNotFragment	143
12.2.1.486	GevSCPSFireTestPacket	143
12.2.1.487	GevSCPSPacketSize	143
12.2.1.488	GevSCSP	143
12.2.1.489	GevSCZoneConfigurationLock	143
12.2.1.490	GevSCZoneCount	143
12.2.1.491	GevSCZoneDirectionAll	143
12.2.1.492	GevSecondURL	143
12.2.1.493	GevStreamChannelSelector	144
12.2.1.494	GevSupportedOption	144
12.2.1.495	GevSupportedOptionSelector	144
12.2.1.496	GevTimestampTickFrequency	144
12.2.1.497	GuiXmlManifestAddress	144
12.2.1.498	Height	144
12.2.1.499	HeightMax	144
12.2.1.500	ImageComponentEnable	144

12.2.1.501 ImageComponentSelector	145
12.2.1.502 ImageCompressionBitrate	145
12.2.1.503 ImageCompressionJPEGFormatOption	145
12.2.1.504 ImageCompressionMode	145
12.2.1.505 ImageCompressionQuality	145
12.2.1.506 ImageCompressionRateOption	145
12.2.1.507 IspEnable	145
12.2.1.508 LineFilterWidth	145
12.2.1.509 LineFormat	146
12.2.1.510 LineInputFilterSelector	146
12.2.1.511 LineInverter	146
12.2.1.512 LineMode	146
12.2.1.513 LinePitch	146
12.2.1.514 LineSelector	146
12.2.1.515 LineSource	146
12.2.1.516 LineStatus	146
12.2.1.517 LineStatusAll	147
12.2.1.518 LinkErrorCount	147
12.2.1.519 LinkUptime	147
12.2.1.520 LogicBlockLUTInputActivation	147
12.2.1.521 LogicBlockLUTInputSelector	147
12.2.1.522 LogicBlockLUTInputSource	147
12.2.1.523 LogicBlockLUTOutputValue	147
12.2.1.524 LogicBlockLUTOutputValueAll	147
12.2.1.525 LogicBlockLUTRowIndex	148
12.2.1.526 LogicBlockLUTSelector	148
12.2.1.527 LogicBlockSelector	148
12.2.1.528 LUTEnable	148
12.2.1.529 LUTIndex	148
12.2.1.530 LUTSelector	148
12.2.1.531 LUTValue	148
12.2.1.532 LUTValueAll	148
12.2.1.533 MaxDeviceResetTime	149
12.2.1.534 OffsetX	149
12.2.1.535 OffsetY	149
12.2.1.536 PacketResendRequestCount	149
12.2.1.537 PayloadSize	149
12.2.1.538 PixelColorFilter	149
12.2.1.539 PixelDynamicRangeMax	149
12.2.1.540 PixelDynamicRangeMin	149
12.2.1.541 PixelFormat	150
12.2.1.542 PixelFormatInfold	150

12.2.1.543 PixelFormatInfoSelector	150
12.2.1.544 PixelSize	150
12.2.1.545 PowerSupplyCurrent	150
12.2.1.546 PowerSupplyVoltage	150
12.2.1.547 RegionDestination	150
12.2.1.548 RegionMode	150
12.2.1.549 RegionSelector	151
12.2.1.550 ReverseX	151
12.2.1.551 ReverseY	151
12.2.1.552 RgbTransformLightSource	151
12.2.1.553 Saturation	151
12.2.1.554 SaturationEnable	151
12.2.1.555 Scan3dAxisMax	151
12.2.1.556 Scan3dAxisMin	151
12.2.1.557 Scan3dCoordinateOffset	152
12.2.1.558 Scan3dCoordinateReferenceSelector	152
12.2.1.559 Scan3dCoordinateReferenceValue	152
12.2.1.560 Scan3dCoordinateScale	152
12.2.1.561 Scan3dCoordinateSelector	152
12.2.1.562 Scan3dCoordinateSystem	152
12.2.1.563 Scan3dCoordinateSystemReference	152
12.2.1.564 Scan3dCoordinateTransformSelector	152
12.2.1.565 Scan3dDistanceUnit	153
12.2.1.566 Scan3dInvalidDataFlag	153
12.2.1.567 Scan3dInvalidDataValue	153
12.2.1.568 Scan3dOutputMode	153
12.2.1.569 Scan3dTransformValue	153
12.2.1.570 SensorDescription	153
12.2.1.571 SensorDigitizationTaps	153
12.2.1.572 SensorHeight	153
12.2.1.573 SensorShutterMode	154
12.2.1.574 SensorTaps	154
12.2.1.575 SensorWidth	154
12.2.1.576 SequencerConfigurationMode	154
12.2.1.577 SequencerConfigurationValid	154
12.2.1.578 SequencerFeatureEnable	154
12.2.1.579 SequencerMode	154
12.2.1.580 SequencerPathSelector	154
12.2.1.581 SequencerSetActive	155
12.2.1.582 SequencerSetLoad	155
12.2.1.583 SequencerSetNext	155
12.2.1.584 SequencerSetSave	155

12.2.1.585 SequencerSetSelector	155
12.2.1.586 SequencerSetStart	155
12.2.1.587 SequencerSetValid	155
12.2.1.588 SequencerTriggerActivation	155
12.2.1.589 SequencerTriggerSource	156
12.2.1.590 SerialPortBaudRate	156
12.2.1.591 SerialPortDataBits	156
12.2.1.592 SerialPortParity	156
12.2.1.593 SerialPortSelector	156
12.2.1.594 SerialPortSource	156
12.2.1.595 SerialPortStopBits	156
12.2.1.596 SerialReceiveFramingErrorCount	156
12.2.1.597 SerialReceiveParityErrorCount	157
12.2.1.598 SerialReceiveQueueClear	157
12.2.1.599 SerialReceiveQueueCurrentCharacterCount	157
12.2.1.600 SerialReceiveQueueMaxCharacterCount	157
12.2.1.601 SerialTransmitQueueCurrentCharacterCount	157
12.2.1.602 SerialTransmitQueueMaxCharacterCount	157
12.2.1.603 Sharpening	157
12.2.1.604 SharpeningAuto	157
12.2.1.605 SharpeningEnable	158
12.2.1.606 SharpeningThreshold	158
12.2.1.607 SoftwareSignalPulse	158
12.2.1.608 SoftwareSignalSelector	158
12.2.1.609 SourceCount	158
12.2.1.610 SourceSelector	158
12.2.1.611 Test0001	158
12.2.1.612 TestEventGenerate	158
12.2.1.613 TestPattern	159
12.2.1.614 TestPatternGeneratorSelector	159
12.2.1.615 TestPendingAck	159
12.2.1.616 TimerDelay	159
12.2.1.617 TimerDuration	159
12.2.1.618 TimerReset	159
12.2.1.619 TimerSelector	159
12.2.1.620 TimerStatus	159
12.2.1.621 TimerTriggerActivation	160
12.2.1.622 TimerTriggerSource	160
12.2.1.623 TimerValue	160
12.2.1.624 Timestamp	160
12.2.1.625 TimestampLatch	160
12.2.1.626 TimestampLatchValue	160

12.2.1.627 TimestampReset	160
12.2.1.628 TLParamsLocked	160
12.2.1.629 TransferAbort	161
12.2.1.630 TransferBlockCount	161
12.2.1.631 TransferBurstCount	161
12.2.1.632 TransferComponentSelector	161
12.2.1.633 TransferControlMode	161
12.2.1.634 TransferOperationMode	161
12.2.1.635 TransferPause	161
12.2.1.636 TransferQueueCurrentBlockCount	161
12.2.1.637 TransferQueueMaxBlockCount	162
12.2.1.638 TransferQueueMode	162
12.2.1.639 TransferQueueOverflowCount	162
12.2.1.640 TransferResume	162
12.2.1.641 TransferSelector	162
12.2.1.642 TransferStart	162
12.2.1.643 TransferStatus	162
12.2.1.644 TransferStatusSelector	162
12.2.1.645 TransferStop	163
12.2.1.646 TransferStreamChannel	163
12.2.1.647 TransferTriggerActivation	163
12.2.1.648 TransferTriggerMode	163
12.2.1.649 TransferTriggerSelector	163
12.2.1.650 TransferTriggerSource	163
12.2.1.651 TriggerActivation	163
12.2.1.652 TriggerDelay	163
12.2.1.653 TriggerDivider	164
12.2.1.654 TriggerEventTest	164
12.2.1.655 TriggerMode	164
12.2.1.656 TriggerMultiplier	164
12.2.1.657 TriggerOverlap	164
12.2.1.658 TriggerSelector	164
12.2.1.659 TriggerSoftware	164
12.2.1.660 TriggerSource	164
12.2.1.661 UserOutputSelector	165
12.2.1.662 UserOutputValue	165
12.2.1.663 UserOutputValueAll	165
12.2.1.664 UserOutputValueAllMask	165
12.2.1.665 UserSetDefault	165
12.2.1.666 UserSetFeatureEnable	165
12.2.1.667 UserSetLoad	165
12.2.1.668 UserSetSave	165

12.2.1.669 UserSetSelector	166
12.2.1.670 V3_3Enable	166
12.2.1.671 WhiteClip	166
12.2.1.672 WhiteClipSelector	166
12.2.1.673 Width	166
12.2.1.674 WidthMax	166
12.3 _quickSpinTLDevice Struct Reference	167
12.3.1 Field Documentation	167
12.3.1.1 DeviceAccessStatus	168
12.3.1.2 DeviceBootloaderVersion	168
12.3.1.3 DeviceCurrentSpeed	168
12.3.1.4 DeviceDisplayName	168
12.3.1.5 DeviceDriverVersion	168
12.3.1.6 DeviceEndianessMechanism	168
12.3.1.7 DeviceID	168
12.3.1.8 DeviceInstanceId	168
12.3.1.9 DevicesUpdater	169
12.3.1.10 DeviceLinkSpeed	169
12.3.1.11 DeviceLocation	169
12.3.1.12 DeviceModelName	169
12.3.1.13 DeviceMulticastMonitorMode	169
12.3.1.14 DevicePortId	169
12.3.1.15 DeviceReset	169
12.3.1.16 DeviceSerialNumber	169
12.3.1.17 DeviceType	170
12.3.1.18 DeviceU3VProtocol	170
12.3.1.19 DeviceUserID	170
12.3.1.20 DeviceVendorName	170
12.3.1.21 DeviceVersion	170
12.3.1.22 GenICamXMLLocation	170
12.3.1.23 GenICamXMLPath	170
12.3.1.24 GevCCP	170
12.3.1.25 GevDeviceAutoForceIP	171
12.3.1.26 GevDeviceDiscoverMaximumPacketSize	171
12.3.1.27 GevDeviceForceGateway	171
12.3.1.28 GevDeviceForceIP	171
12.3.1.29 GevDeviceForceIPAddress	171
12.3.1.30 GevDeviceForceSubnetMask	171
12.3.1.31 GevDeviceGateway	171
12.3.1.32 GevDeviceIPAddress	171
12.3.1.33 GevDevicesWrongSubnet	172
12.3.1.34 GevDeviceMACAddress	172

12.3.1.35	GevDeviceMaximumPacketSize	172
12.3.1.36	GevDeviceMaximumRetryCount	172
12.3.1.37	GevDeviceModelsBigEndian	172
12.3.1.38	GevDevicePort	172
12.3.1.39	GevDeviceReadAndWriteTimeout	172
12.3.1.40	GevDeviceSubnetMask	172
12.3.1.41	GevVersionMajor	173
12.3.1.42	GevVersionMinor	173
12.3.1.43	GUIXMLLocation	173
12.3.1.44	GUIXMLPath	173
12.4	_quickSpinTLInterface Struct Reference	173
12.4.1	Field Documentation	174
12.4.1.1	ActionCommand	174
12.4.1.2	DeviceAccessStatus	174
12.4.1.3	DeviceCount	175
12.4.1.4	DeviceID	175
12.4.1.5	DeviceModelName	175
12.4.1.6	DeviceSelector	175
12.4.1.7	DeviceSerialNumber	175
12.4.1.8	DeviceUnlock	175
12.4.1.9	DeviceUpdateList	175
12.4.1.10	DeviceVendorName	175
12.4.1.11	FilterDriverStatus	176
12.4.1.12	GevActionDeviceKey	176
12.4.1.13	GevActionGroupKey	176
12.4.1.14	GevActionGroupMask	176
12.4.1.15	GevActionTime	176
12.4.1.16	GevDeviceAutoForceIP	176
12.4.1.17	GevDeviceForceGateway	176
12.4.1.18	GevDeviceForceIP	176
12.4.1.19	GevDeviceForceIPAddress	177
12.4.1.20	GevDeviceForceSubnetMask	177
12.4.1.21	GevDeviceGateway	177
12.4.1.22	GevDeviceIPAddress	177
12.4.1.23	GevDeviceMACAddress	177
12.4.1.24	GevDeviceSubnetMask	177
12.4.1.25	GevInterfaceGateway	177
12.4.1.26	GevInterfaceGatewaySelector	177
12.4.1.27	GevInterfaceMACAddress	178
12.4.1.28	GevInterfaceMTU	178
12.4.1.29	GevInterfaceReceiveLinkSpeed	178
12.4.1.30	GevInterfaceSubnetIPAddress	178

12.4.1.31	GevInterfaceSubnetMask	178
12.4.1.32	GevInterfaceSubnetSelector	178
12.4.1.33	GevInterfaceTransmitLinkSpeed	178
12.4.1.34	HostAdapterDriverVersion	178
12.4.1.35	HostAdapterName	179
12.4.1.36	HostAdapterVendor	179
12.4.1.37	IncompatibleDeviceCount	179
12.4.1.38	IncompatibleDeviceID	179
12.4.1.39	IncompatibleDeviceModelName	179
12.4.1.40	IncompatibleDeviceSelector	179
12.4.1.41	IncompatibleDeviceVendorName	179
12.4.1.42	IncompatibleGevDeviceIPAddress	179
12.4.1.43	IncompatibleGevDeviceMACAddress	180
12.4.1.44	IncompatibleGevDeviceSubnetMask	180
12.4.1.45	InterfaceDisplayName	180
12.4.1.46	InterfaceID	180
12.4.1.47	InterfaceType	180
12.4.1.48	POEStatus	180
12.5	_quickSpinTLStream Struct Reference	181
12.5.1	Field Documentation	181
12.5.1.1	StreamAnnounceBufferMinimum	181
12.5.1.2	StreamAnnouncedBufferCount	181
12.5.1.3	StreamBlockTransferSize	182
12.5.1.4	StreamBufferAlignment	182
12.5.1.5	StreamBufferCountManual	182
12.5.1.6	StreamBufferCountMax	182
12.5.1.7	StreamBufferCountMode	182
12.5.1.8	StreamBufferCountResult	182
12.5.1.9	StreamBufferHandlingMode	182
12.5.1.10	StreamChunkCountMaximum	182
12.5.1.11	StreamCRCCheckEnable	183
12.5.1.12	StreamDeliveredFrameCount	183
12.5.1.13	StreamDroppedFrameCount	183
12.5.1.14	StreamID	183
12.5.1.15	StreamIncompleteFrameCount	183
12.5.1.16	StreamInputBufferCount	183
12.5.1.17	StreamIsGrabbing	183
12.5.1.18	StreamLostFrameCount	183
12.5.1.19	StreamMissedPacketCount	184
12.5.1.20	StreamMode	184
12.5.1.21	StreamOutputBufferCount	184
12.5.1.22	StreamPacketResendEnable	184

12.5.1.23 StreamPacketResendMaxRequests	184
12.5.1.24 StreamPacketResendReceivedPacketCount	184
12.5.1.25 StreamPacketResendRequestCount	184
12.5.1.26 StreamPacketResendRequestedPacketCount	184
12.5.1.27 StreamPacketResendRequestSuccessCount	185
12.5.1.28 StreamPacketResendTimeout	185
12.5.1.29 StreamReceivedFrameCount	185
12.5.1.30 StreamReceivedPacketCount	185
12.5.1.31 StreamStartedFrameCount	185
12.5.1.32 StreamType	185
12.6 _quickSpinTLSysSystem Struct Reference	186
12.6.1 Field Documentation	186
12.6.1.1 EnumerateGen2Cameras	186
12.6.1.2 EnumerateGEVInterfaces	186
12.6.1.3 EnumerateUSBInterfaces	187
12.6.1.4 GenTLFNCVersionMajor	187
12.6.1.5 GenTLFNCVersionMinor	187
12.6.1.6 GenTLFNCVersionSubMinor	187
12.6.1.7 GenTLVersionMajor	187
12.6.1.8 GenTLVersionMinor	187
12.6.1.9 GevInterfaceDefaultGateway	187
12.6.1.10 GevInterfaceDefaultIPAddress	187
12.6.1.11 GevInterfaceDefaultSubnetMask	188
12.6.1.12 GevInterfaceMACAddress	188
12.6.1.13 GevVersionMajor	188
12.6.1.14 GevVersionMinor	188
12.6.1.15 InterfaceDisplayName	188
12.6.1.16 InterfaceID	188
12.6.1.17 InterfaceSelector	188
12.6.1.18 InterfaceUpdateList	188
12.6.1.19 TLDisplayName	189
12.6.1.20 TLFileName	189
12.6.1.21 TLID	189
12.6.1.22 TLModelName	189
12.6.1.23 TLPath	189
12.6.1.24 TLType	189
12.6.1.25 TLVendorName	189
12.6.1.26 TLVersion	190
12.7 _spinAVIOption Struct Reference	190
12.7.1 Detailed Description	190
12.7.2 Field Documentation	190
12.7.2.1 frameRate	190

12.7.2.2 height	190
12.7.2.3 reserved	191
12.7.2.4 width	191
12.8 _spinBMPOption Struct Reference	191
12.8.1 Detailed Description	191
12.8.2 Field Documentation	191
12.8.2.1 indexedColor_8bit	191
12.8.2.2 reserved	192
12.9 _spinChunkData Struct Reference	192
12.9.1 Detailed Description	193
12.9.2 Field Documentation	193
12.9.2.1 m_blackLevel	193
12.9.2.2 m_compressionMode	193
12.9.2.3 m_compressionRatio	193
12.9.2.4 m_counterValue	193
12.9.2.5 m_cRC	193
12.9.2.6 m_encoderValue	193
12.9.2.7 m_exposureEndLineStatusAll	194
12.9.2.8 m_exposureTime	194
12.9.2.9 m_frameID	194
12.9.2.10 m_gain	194
12.9.2.11 m_height	194
12.9.2.12 m_image	194
12.9.2.13 m_inferenceConfidence	194
12.9.2.14 m_inferenceFrameId	194
12.9.2.15 m_inferenceResult	195
12.9.2.16 m_linePitch	195
12.9.2.17 m_lineStatusAll	195
12.9.2.18 m_offsetX	195
12.9.2.19 m_offsetY	195
12.9.2.20 m_partSelector	195
12.9.2.21 m_pixelDynamicRangeMax	195
12.9.2.22 m_pixelDynamicRangeMin	195
12.9.2.23 m_scan3dAxisMax	196
12.9.2.24 m_scan3dAxisMin	196
12.9.2.25 m_scan3dCoordinateOffset	196
12.9.2.26 m_scan3dCoordinateReferenceValue	196
12.9.2.27 m_scan3dCoordinateScale	196
12.9.2.28 m_scan3dInvalidDataValue	196
12.9.2.29 m_scan3dTransformValue	196
12.9.2.30 m_scanLineSelector	196
12.9.2.31 m_sequencerSetActive	197

12.9.2.32 m_serialDataLength	197
12.9.2.33 m_streamChannelID	197
12.9.2.34 m_timerValue	197
12.9.2.35 m_timestamp	197
12.9.2.36 m_timestampLatchValue	197
12.9.2.37 m_transferBlockID	197
12.9.2.38 m_transferQueueCurrentBlockCount	197
12.9.2.39 m_width	198
12.10 _spinH264Option Struct Reference	198
12.10.1 Detailed Description	198
12.10.2 Field Documentation	198
12.10.2.1 bitrate	198
12.10.2.2 frameRate	199
12.10.2.3 height	199
12.10.2.4 reserved	199
12.10.2.5 width	199
12.11 _spinJPEGOption Struct Reference	199
12.11.1 Detailed Description	200
12.11.2 Field Documentation	200
12.11.2.1 progressive	200
12.11.2.2 quality	200
12.11.2.3 reserved	200
12.12 _spinJPG2Option Struct Reference	200
12.12.1 Detailed Description	201
12.12.2 Field Documentation	201
12.12.2.1 quality	201
12.12.2.2 reserved	201
12.13 _spinLibraryVersion Struct Reference	201
12.13.1 Detailed Description	202
12.13.2 Field Documentation	202
12.13.2.1 build	202
12.13.2.2 major	202
12.13.2.3 minor	202
12.13.2.4 type	202
12.14 _spinMJPGOption Struct Reference	202
12.14.1 Detailed Description	203
12.14.2 Field Documentation	203
12.14.2.1 frameRate	203
12.14.2.2 height	203
12.14.2.3 quality	203
12.14.2.4 reserved	204
12.14.2.5 width	204

12.15 _spinPGMOption Struct Reference	204
12.15.1 Detailed Description	204
12.15.2 Field Documentation	204
12.15.2.1 binaryFile	204
12.15.2.2 reserved	205
12.16 _spinPNGOption Struct Reference	205
12.16.1 Detailed Description	205
12.16.2 Field Documentation	205
12.16.2.1 compressionLevel	205
12.16.2.2 interlaced	205
12.16.2.3 reserved	206
12.17 _spinPPMOption Struct Reference	206
12.17.1 Detailed Description	206
12.17.2 Field Documentation	206
12.17.2.1 binaryFile	206
12.17.2.2 reserved	206
12.18 _spinTIFFOption Struct Reference	207
12.18.1 Detailed Description	207
12.18.2 Field Documentation	207
12.18.2.1 compression	207
12.18.2.2 reserved	207
13 File Documentation	209
13.1 doc/spindocs/C/GettingStarted.dox File Reference	209
13.2 doc/spindocs/C/ProgrammerGuide.dox File Reference	209
13.3 doc/spindocs/shared/Benefits.dox File Reference	209
13.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference	209
13.5 doc/spindocs/shared/GenICamGenTL.dox File Reference	209
13.6 doc/spindocs/shared/Licensing.dox File Reference	209
13.7 doc/spindocs/shared/Maintenance.dox File Reference	209
13.8 include/spinc/CameraDefsC.h File Reference	209
13.8.1 Enumeration Type Documentation	242
13.8.1.1 _spinAcquisitionModeEnums	242
13.8.1.2 _spinAcquisitionStatusSelectorEnums	242
13.8.1.3 _spinActionUnconditionalModeEnums	243
13.8.1.4 _spinAdcBitDepthEnums	243
13.8.1.5 _spinAutoAlgorithmSelectorEnums	243
13.8.1.6 _spinAutoExposureControlPriorityEnums	244
13.8.1.7 _spinAutoExposureLightingModeEnums	244
13.8.1.8 _spinAutoExposureMeteringModeEnums	244
13.8.1.9 _spinAutoExposureTargetGreyValueAutoEnums	245
13.8.1.10 _spinBalanceRatioSelectorEnums	245

13.8.1.11 _spinBalanceWhiteAutoEnums	245
13.8.1.12 _spinBalanceWhiteAutoProfileEnums	246
13.8.1.13 _spinBinningHorizontalModeEnums	246
13.8.1.14 _spinBinningSelectorEnums	247
13.8.1.15 _spinBinningVerticalModeEnums	247
13.8.1.16 _spinBlackLevelAutoBalanceEnums	247
13.8.1.17 _spinBlackLevelAutoEnums	248
13.8.1.18 _spinBlackLevelSelectorEnums	248
13.8.1.19 _spinChunkBlackLevelSelectorEnums	248
13.8.1.20 _spinChunkCounterSelectorEnums	248
13.8.1.21 _spinChunkEncoderSelectorEnums	249
13.8.1.22 _spinChunkEncoderStatusEnums	249
13.8.1.23 _spinChunkExposureTimeSelectorEnums	249
13.8.1.24 _spinChunkGainSelectorEnums	250
13.8.1.25 _spinChunkImageComponentEnums	250
13.8.1.26 _spinChunkPixelFormatEnums	251
13.8.1.27 _spinChunkRegionIDEnums	251
13.8.1.28 _spinChunkScan3dCoordinateReferenceSelectorEnums	251
13.8.1.29 _spinChunkScan3dCoordinateSelectorEnums	252
13.8.1.30 _spinChunkScan3dCoordinateSystemEnums	252
13.8.1.31 _spinChunkScan3dCoordinateSystemReferenceEnums	252
13.8.1.32 _spinChunkScan3dCoordinateTransformSelectorEnums	253
13.8.1.33 _spinChunkScan3dDistanceUnitEnums	253
13.8.1.34 _spinChunkScan3dOutputModeEnums	254
13.8.1.35 _spinChunkSelectorEnums	255
13.8.1.36 _spinChunkSourceIDEnums	255
13.8.1.37 _spinChunkTimerSelectorEnums	255
13.8.1.38 _spinChunkTransferStreamIDEnums	256
13.8.1.39 _spinCIConfigurationEnums	256
13.8.1.40 _spinCITimeSlotsCountEnums	257
13.8.1.41 _spinColorTransformationSelectorEnums	257
13.8.1.42 _spinColorTransformationValueSelectorEnums	257
13.8.1.43 _spinCompressionSaturationPriorityEnums	258
13.8.1.44 _spinCounterEventActivationEnums	258
13.8.1.45 _spinCounterEventSourceEnums	258
13.8.1.46 _spinCounterResetActivationEnums	259
13.8.1.47 _spinCounterResetSourceEnums	259
13.8.1.48 _spinCounterSelectorEnums	260
13.8.1.49 _spinCounterStatusEnums	260
13.8.1.50 _spinCounterTriggerActivationEnums	261
13.8.1.51 _spinCounterTriggerSourceEnums	261
13.8.1.52 _spinCxpConnectionTestModeEnums	262

13.8.1.53 _spinCxpLinkConfigurationEnums	262
13.8.1.54 _spinCxpLinkConfigurationPreferredEnums	263
13.8.1.55 _spinCxpLinkConfigurationStatusEnums	264
13.8.1.56 _spinCxpPoCxpStatusEnums	265
13.8.1.57 _spinDecimationHorizontalModeEnums	265
13.8.1.58 _spinDecimationSelectorEnums	265
13.8.1.59 _spinDecimationVerticalModeEnums	266
13.8.1.60 _spinDefectCorrectionModeEnums	266
13.8.1.61 _spinDeinterlacingEnums	266
13.8.1.62 _spinDeviceCharacterSetEnums	267
13.8.1.63 _spinDeviceClockSelectorEnums	267
13.8.1.64 _spinDeviceConnectionStatusEnums	267
13.8.1.65 _spinDeviceIndicatorModeEnums	268
13.8.1.66 _spinDeviceLinkHeartbeatModeEnums	268
13.8.1.67 _spinDeviceLinkThroughputLimitModeEnums	268
13.8.1.68 _spinDevicePowerSupplySelectorEnums	269
13.8.1.69 _spinDeviceRegistersEndiannessEnums	269
13.8.1.70 _spinDeviceScanTypeEnums	269
13.8.1.71 _spinDeviceSerialPortBaudRateEnums	269
13.8.1.72 _spinDeviceSerialPortSelectorEnums	270
13.8.1.73 _spinDeviceStreamChannelEndiannessEnums	270
13.8.1.74 _spinDeviceStreamChannelTypeEnums	270
13.8.1.75 _spinDeviceTapGeometryEnums	271
13.8.1.76 _spinDeviceTemperatureSelectorEnums	272
13.8.1.77 _spinDeviceTLTypeEnums	272
13.8.1.78 _spinDeviceTypeEnums	273
13.8.1.79 _spinEncoderModeEnums	273
13.8.1.80 _spinEncoderOutputModeEnums	273
13.8.1.81 _spinEncoderResetActivationEnums	274
13.8.1.82 _spinEncoderResetSourceEnums	274
13.8.1.83 _spinEncoderSelectorEnums	275
13.8.1.84 _spinEncoderSourceAEnums	276
13.8.1.85 _spinEncoderSourceBEnums	276
13.8.1.86 _spinEncoderStatusEnums	276
13.8.1.87 _spinEventNotificationEnums	277
13.8.1.88 _spinEventSelectorEnums	277
13.8.1.89 _spinExposureActiveModeEnums	277
13.8.1.90 _spinExposureAutoEnums	278
13.8.1.91 _spinExposureModeEnums	278
13.8.1.92 _spinExposureTimeModeEnums	278
13.8.1.93 _spinExposureTimeSelectorEnums	279
13.8.1.94 _spinFileOpenModeEnums	279

13.8.1.95 _spinFileOperationSelectorEnums	279
13.8.1.96 _spinFileOperationStatusEnums	280
13.8.1.97 _spinFileSelectorEnums	280
13.8.1.98 _spinGainAutoBalanceEnums	280
13.8.1.99 _spinGainAutoEnums	282
13.8.1.100 _spinGainSelectorEnums	282
13.8.1.101 _spinGevCCPEnums	282
13.8.1.102 _spinGevCurrentPhysicalLinkConfigurationEnums	283
13.8.1.103 _spinGevGVCPExtendedStatusCodesSelectorEnums	283
13.8.1.104 _spinGevGVSPExtendedIDModeEnums	283
13.8.1.105 _spinGevIEEE1588ClockAccuracyEnums	284
13.8.1.106 _spinGevIEEE1588ModeEnums	284
13.8.1.107 _spinGevIEEE1588StatusEnums	284
13.8.1.108 _spinGevIPCConfigurationStatusEnums	285
13.8.1.109 _spinGevPhysicalLinkConfigurationEnums	285
13.8.1.110 _spinGevSupportedOptionSelectorEnums	286
13.8.1.111 _spinImageComponentSelectorEnums	286
13.8.1.112 _spinImageCompressionJPEGFormatOptionEnums	287
13.8.1.113 _spinImageCompressionModeEnums	287
13.8.1.114 _spinImageCompressionRateOptionEnums	288
13.8.1.115 _spinLineFormatEnums	288
13.8.1.116 _spinLineInputFilterSelectorEnums	288
13.8.1.117 _spinLineModeEnums	289
13.8.1.118 _spinLineSelectorEnums	289
13.8.1.119 _spinLineSourceEnums	289
13.8.1.120 _spinLogicBlockLUTInputActivationEnums	290
13.8.1.121 _spinLogicBlockLUTInputSelectorEnums	290
13.8.1.122 _spinLogicBlockLUTInputSourceEnums	291
13.8.1.123 _spinLogicBlockLUTSelectorEnums	291
13.8.1.124 _spinLogicBlockSelectorEnums	292
13.8.1.125 _spinLUTSelectorEnums	292
13.8.1.126 _spinPixelColorFilterEnums	292
13.8.1.127 _spinPixelFormatEnums	293
13.8.1.128 _spinPixelFormatInfoSelectorEnums	298
13.8.1.129 _spinPixelSizeEnums	304
13.8.1.130 _spinRegionDestinationEnums	305
13.8.1.131 _spinRegionModeEnums	305
13.8.1.132 _spinRegionSelectorEnums	305
13.8.1.133 _spinRgbTransformLightSourceEnums	306
13.8.1.134 _spinScan3dCoordinateReferenceSelectorEnums	306
13.8.1.135 _spinScan3dCoordinateSelectorEnums	307
13.8.1.136 _spinScan3dCoordinateSystemEnums	307

13.8.1.137 _spinScan3dCoordinateSystemReferenceEnums	307
13.8.1.138 _spinScan3dCoordinateTransformSelectorEnums	308
13.8.1.139 _spinScan3dDistanceUnitEnums	308
13.8.1.140 _spinScan3dOutputModeEnums	308
13.8.1.141 _spinSensorDigitizationTapsEnums	309
13.8.1.142 _spinSensorShutterModeEnums	310
13.8.1.143 _spinSensorTapsEnums	310
13.8.1.144 _spinSequencerConfigurationModeEnums	311
13.8.1.145 _spinSequencerConfigurationValidEnums	311
13.8.1.146 _spinSequencerModeEnums	311
13.8.1.147 _spinSequencerSetValidEnums	311
13.8.1.148 _spinSequencerTriggerActivationEnums	312
13.8.1.149 _spinSequencerTriggerSourceEnums	312
13.8.1.150 _spinSerialPortBaudRateEnums	312
13.8.1.151 _spinSerialPortParityEnums	313
13.8.1.152 _spinSerialPortSelectorEnums	313
13.8.1.153 _spinSerialPortSourceEnums	314
13.8.1.154 _spinSerialPortStopBitsEnums	314
13.8.1.155 _spinSoftwareSignalSelectorEnums	314
13.8.1.156 _spinSourceSelectorEnums	315
13.8.1.157 _spinTestPatternEnums	315
13.8.1.158 _spinTestPatternGeneratorSelectorEnums	315
13.8.1.159 _spinTimerSelectorEnums	316
13.8.1.160 _spinTimerStatusEnums	316
13.8.1.161 _spinTimerTriggerActivationEnums	316
13.8.1.162 _spinTimerTriggerSourceEnums	317
13.8.1.163 _spinTransferComponentSelectorEnums	318
13.8.1.164 _spinTransferControlModeEnums	318
13.8.1.165 _spinTransferOperationModeEnums	319
13.8.1.166 _spinTransferQueueModeEnums	319
13.8.1.167 _spinTransferSelectorEnums	319
13.8.1.168 _spinTransferStatusSelectorEnums	320
13.8.1.169 _spinTransferTriggerActivationEnums	320
13.8.1.170 _spinTransferTriggerModeEnums	320
13.8.1.171 _spinTransferTriggerSelectorEnums	321
13.8.1.172 _spinTransferTriggerSourceEnums	321
13.8.1.173 _spinTriggerActivationEnums	322
13.8.1.174 _spinTriggerModeEnums	323
13.8.1.175 _spinTriggerOverlapEnums	323
13.8.1.176 _spinTriggerSelectorEnums	323
13.8.1.177 _spinTriggerSourceEnums	324
13.8.1.178 _spinUserOutputSelectorEnums	324

13.8.1.179 _spinUserSetDefaultEnums	324
13.8.1.180 _spinUserSetSelectorEnums	325
13.8.1.181 _spinWhiteClipSelectorEnums	325
13.9 include/spinc/ChunkDataDefC.h File Reference	326
13.10 include/spinc/QuickSpinC.h File Reference	327
13.10.1 Function Documentation	327
13.10.1.1 quickSpinInit()	327
13.10.1.2 quickSpinInitEx()	328
13.10.1.3 quickSpinTLDeviceInit()	328
13.10.1.4 quickSpinTLInterfaceInit()	328
13.10.1.5 quickSpinTLStreamInit()	328
13.10.1.6 quickSpinTLSystemInit()	328
13.11 include/spinc/QuickSpinDefsC.h File Reference	329
13.11.1 Typedef Documentation	329
13.11.1.1 quickSpinBooleanNode	330
13.11.1.2 quickSpinCommandNode	330
13.11.1.3 quickSpinEnumerationNode	330
13.11.1.4 quickSpinFloatNode	330
13.11.1.5 quickSpinIntegerNode	330
13.11.1.6 quickSpinRegisterNode	330
13.11.1.7 quickSpinStringNode	330
13.12 include/spinc/SpinnakerC.h File Reference	331
13.12.1 Function Documentation	340
13.12.1.1 spinCameraBeginAcquisition()	341
13.12.1.2 spinCameraDelInit()	341
13.12.1.3 spinCameraDiscoverMaxPacketSize()	341
13.12.1.4 spinCameraEndAcquisition()	342
13.12.1.5 spinCameraForceIP()	342
13.12.1.6 spinCameraGetAccessMode()	343
13.12.1.7 spinCameraGetGuiXml()	343
13.12.1.8 spinCameraGetNextImage()	344
13.12.1.9 spinCameraGetNextImageEx()	344
13.12.1.10 spinCameraGetNextImageSync()	345
13.12.1.11 spinCameraGetNodeMap()	345
13.12.1.12 spinCameraGetTLDeviceNodeMap()	346
13.12.1.13 spinCameraGetTLStreamNodeMap()	346
13.12.1.14 spinCameraGetUniqueID()	347
13.12.1.15 spinCameraInit()	347
13.12.1.16 spinCameraIsInitialized()	348
13.12.1.17 spinCameraIsStreaming()	348
13.12.1.18 spinCameraIsValid()	348
13.12.1.19 spinCameraListAppend()	349

13.12.1.20 spinCameraListClear()	349
13.12.1.21 spinCameraListCreateEmpty()	350
13.12.1.22 spinCameraListDestroy()	350
13.12.1.23 spinCameraListGet()	351
13.12.1.24 spinCameraListGetBySerial()	351
13.12.1.25 spinCameraListGetSize()	352
13.12.1.26 spinCameraListRemove()	352
13.12.1.27 spinCameraListRemoveBySerial()	353
13.12.1.28 spinCameraReadPort()	353
13.12.1.29 spinCameraRegisterDeviceEventHandler()	354
13.12.1.30 spinCameraRegisterDeviceEventHandlerEx()	354
13.12.1.31 spinCameraRegisterImageEventHandler()	355
13.12.1.32 spinCameraRegisterImageEventHandlerEx()	355
13.12.1.33 spinCameraRegisterImageListEventHandler()	356
13.12.1.34 spinCameraRelease()	356
13.12.1.35 spinCameraUnregisterDeviceEventHandler()	356
13.12.1.36 spinCameraUnregisterImageEventHandler()	357
13.12.1.37 spinCameraUnregisterImageListEventHandler()	357
13.12.1.38 spinCameraWritePort()	358
13.12.1.39 spinDeviceArrivalEventHandlerCreate()	358
13.12.1.40 spinDeviceArrivalEventHandlerDestroy()	359
13.12.1.41 spinDeviceEventGetId()	359
13.12.1.42 spinDeviceEventGetName()	359
13.12.1.43 spinDeviceEventGetPayloadData()	360
13.12.1.44 spinDeviceEventGetPayloadDataSize()	361
13.12.1.45 spinDeviceEventHandlerCreate()	361
13.12.1.46 spinDeviceEventHandlerDestroy()	362
13.12.1.47 spinDeviceRemovalEventHandlerCreate()	362
13.12.1.48 spinDeviceRemovalEventHandlerDestroy()	363
13.12.1.49 spinErrorGetLast()	363
13.12.1.50 spinErrorGetLastBuildDate()	363
13.12.1.51 spinErrorGetLastBuildTime()	364
13.12.1.52 spinErrorGetLastFileName()	364
13.12.1.53 spinErrorGetLastFullMessage()	365
13.12.1.54 spinErrorGetLastFunctionName()	365
13.12.1.55 spinErrorGetLastLineNumber()	366
13.12.1.56 spinErrorGetLastMessage()	366
13.12.1.57 spinImageCalculateStatistics()	367
13.12.1.58 spinImageCheckCRC()	367
13.12.1.59 spinImageChunkDataGetFloatValue()	368
13.12.1.60 spinImageChunkDataGetIntValue()	368
13.12.1.61 spinImageCreate()	368

13.12.1.62 spinImageCreateEmpty()	369
13.12.1.63 spinImageCreateEx()	369
13.12.1.64 spinImageCreateEx2()	370
13.12.1.65 spinImageDeepCopy()	370
13.12.1.66 spinImageDestroy()	371
13.12.1.67 spinImageEventHandlerCreate()	371
13.12.1.68 spinImageEventHandlerDestroy()	372
13.12.1.69 spinImageGetBitsPerPixel()	372
13.12.1.70 spinImageGetBufferSize()	373
13.12.1.71 spinImageGetChunkLayoutID()	373
13.12.1.72 spinImageGetColorProcessing()	374
13.12.1.73 spinImageGetData()	374
13.12.1.74 spinImageGetFrameID()	375
13.12.1.75 spinImageGetHeight()	375
13.12.1.76 spinImageGetID()	376
13.12.1.77 spinImageGetOffsetX()	376
13.12.1.78 spinImageGetOffsetY()	377
13.12.1.79 spinImageGetPaddingX()	377
13.12.1.80 spinImageGetPaddingY()	378
13.12.1.81 spinImageGetPayloadType()	378
13.12.1.82 spinImageGetPixelFormat()	379
13.12.1.83 spinImageGetPixelFormatName()	379
13.12.1.84 spinImageGetPrivateData()	380
13.12.1.85 spinImageGetSize()	380
13.12.1.86 spinImageGetStatus()	381
13.12.1.87 spinImageGetStatusDescription()	381
13.12.1.88 spinImageGetStride()	382
13.12.1.89 spinImageGetTimeStamp()	382
13.12.1.90 spinImageGetTLPayloadType()	383
13.12.1.91 spinImageGetTLPixelFormat()	383
13.12.1.92 spinImageGetTLPixelFormatNamespace()	384
13.12.1.93 spinImageGetValidPayloadSize()	384
13.12.1.94 spinImageGetWidth()	385
13.12.1.95 spinImageHasCRC()	385
13.12.1.96 spinImageIsIncomplete()	386
13.12.1.97 spinImageListAppend()	386
13.12.1.98 spinImageListClear()	387
13.12.1.99 spinImageListCreateEmpty()	387
13.12.1.100 spinImageListDestroy()	388
13.12.1.101 spinImageListEventHandlerCreate()	388
13.12.1.102 spinImageListEventHandlerDestroy()	389
13.12.1.103 spinImageListGet()	389

13.12.1.104 spinImageListGetByPixelFormat()	390
13.12.1.105 spinImageListGetSize()	390
13.12.1.106 spinImageListLoad()	391
13.12.1.107 spinImageListRelease()	391
13.12.1.108 spinImageListRemove()	391
13.12.1.109 spinImageListRemoveByPixelFormat()	392
13.12.1.110 spinImageListSave()	392
13.12.1.111 spinImageProcessorApplyGamma()	393
13.12.1.112 spinImageProcessorConvert()	393
13.12.1.113 spinImageProcessorConvertImageList()	394
13.12.1.114 spinImageProcessorCreate()	395
13.12.1.115 spinImageProcessorDestroy()	395
13.12.1.116 spinImageProcessorGetColorProcessing()	396
13.12.1.117 spinImageProcessorGetNumDecompressionThreads()	396
13.12.1.118 spinImageProcessorSetColorProcessing()	396
13.12.1.119 spinImageProcessorSetNumDecompressionThreads()	397
13.12.1.120 spinImageRelease()	397
13.12.1.121 spinImageReset()	398
13.12.1.122 spinImageResetEx()	398
13.12.1.123 spinImageSave()	399
13.12.1.124 spinImageSaveBmp()	400
13.12.1.125 spinImageSaveFromExt()	400
13.12.1.126 spinImageSaveJpeg()	401
13.12.1.127 spinImageSaveJpg2()	401
13.12.1.128 spinImageSavePgm()	402
13.12.1.129 spinImageSavePng()	402
13.12.1.130 spinImageSavePpm()	403
13.12.1.131 spinImageSaveTiff()	403
13.12.1.132 spinImageStatisticsCreate()	404
13.12.1.133 spinImageStatisticsDestroy()	404
13.12.1.134 spinImageStatisticsDisableAll()	404
13.12.1.135 spinImageStatisticsEnableAll()	405
13.12.1.136 spinImageStatisticsEnableGreyOnly()	405
13.12.1.137 spinImageStatisticsEnableHslOnly()	406
13.12.1.138 spinImageStatisticsEnableRgbOnly()	406
13.12.1.139 spinImageStatisticsGetAll()	407
13.12.1.140 spinImageStatisticsGetChannelStatus()	407
13.12.1.141 spinImageStatisticsGetHistogram()	408
13.12.1.142 spinImageStatisticsGetMean()	408
13.12.1.143 spinImageStatisticsGetNumPixelValues()	409
13.12.1.144 spinImageStatisticsGetPixelValueRange()	409
13.12.1.145 spinImageStatisticsGetRange()	410

13.12.1.146 spinImageStatisticsSetChannelStatus()	410
13.12.1.147 spinInterfaceEventHandlerCreate()	411
13.12.1.148 spinInterfaceEventHandlerDestroy()	411
13.12.1.149 spinInterfaceGetCameras()	412
13.12.1.150 spinInterfaceGetCamerasEx()	412
13.12.1.151 spinInterfaceGetTLNodeMap()	413
13.12.1.152 spinInterfaceIsInUse()	413
13.12.1.153 spinInterfaceListClear()	414
13.12.1.154 spinInterfaceListCreateEmpty()	414
13.12.1.155 spinInterfaceListDestroy()	415
13.12.1.156 spinInterfaceListGet()	415
13.12.1.157 spinInterfaceListGetSize()	416
13.12.1.158 spinInterfaceRegisterDeviceArrivalEventHandler()	416
13.12.1.159 spinInterfaceRegisterDeviceRemovalEventHandler()	417
13.12.1.160 spinInterfaceRegisterInterfaceEventHandler()	417
13.12.1.161 spinInterfaceRelease()	418
13.12.1.162 spinInterfaceSendActionCommand()	418
13.12.1.163 spinInterfaceUnregisterDeviceArrivalEventHandler()	419
13.12.1.164 spinInterfaceUnregisterDeviceRemovalEventHandler()	419
13.12.1.165 spinInterfaceUnregisterInterfaceEventHandler()	420
13.12.1.166 spinInterfaceUpdateCameras()	420
13.12.1.167 spinLogDataGetCategoryName()	421
13.12.1.168 spinLogDataGetLogMessage()	421
13.12.1.169 spinLogDataGetNDC()	422
13.12.1.170 spinLogDataGetPriority()	422
13.12.1.171 spinLogDataGetPriorityName()	423
13.12.1.172 spinLogDataGetThreadName()	423
13.12.1.173 spinLogDataGetTimestamp()	424
13.12.1.174 spinLogEventHandlerCreate()	424
13.12.1.175 spinLogEventHandlerDestroy()	425
13.12.1.176 spinSystemGetCameras()	425
13.12.1.177 spinSystemGetCamerasEx()	426
13.12.1.178 spinSystemGetInstance()	426
13.12.1.179 spinSystemGetInterfaces()	427
13.12.1.180 spinSystemGetLibraryVersion()	427
13.12.1.181 spinSystemGetLoggingLevel()	428
13.12.1.182 spinSystemGetTLNodeMap()	428
13.12.1.183 spinSystemIsInUse()	429
13.12.1.184 spinSystemRegisterDeviceArrivalEventHandler()	429
13.12.1.185 spinSystemRegisterDeviceRemovalEventHandler()	430
13.12.1.186 spinSystemRegisterInterfaceEventHandler()	430
13.12.1.187 spinSystemRegisterLogEventHandler()	431

13.12.1.188 spinSystemReleaseInstance()	431
13.12.1.189 spinSystemSendActionCommand()	431
13.12.1.190 spinSystemSetLoggingLevel()	432
13.12.1.191 spinSystemUnregisterAllLogEventHandlers()	433
13.12.1.192 spinSystemUnregisterDeviceArrivalEventHandler()	433
13.12.1.193 spinSystemUnregisterDeviceRemovalEventHandler()	434
13.12.1.194 spinSystemUnregisterInterfaceEventHandler()	434
13.12.1.195 spinSystemUnregisterLogEventHandler()	435
13.12.1.196 spinSystemUpdateCameras()	435
13.12.1.197 spinSystemUpdateCamerasEx()	436
13.13 include/spinc/SpinnakerDefsC.h File Reference	436
13.13.1 Typedef Documentation	441
13.13.1.1 bool8_t	441
13.13.1.2 spinArrivalEventFunction	441
13.13.1.3 spinCamera	441
13.13.1.4 spinCameraList	442
13.13.1.5 spinDeviceArrivalEventHandler	442
13.13.1.6 spinDeviceEventData	442
13.13.1.7 spinDeviceEventFunction	442
13.13.1.8 spinDeviceEventHandler	442
13.13.1.9 spinDeviceRemovalEventHandler	442
13.13.1.10 spinImage	443
13.13.1.11 spinImageEventFunction	443
13.13.1.12 spinImageEventHandler	443
13.13.1.13 spinImageList	443
13.13.1.14 spinImageListEventFunction	443
13.13.1.15 spinImageListEventHandler	443
13.13.1.16 spinImageProcessor	444
13.13.1.17 spinImageStatistics	444
13.13.1.18 spinInterface	444
13.13.1.19 spinInterfaceEventHandler	444
13.13.1.20 spinInterfaceList	444
13.13.1.21 spinLogEventData	444
13.13.1.22 spinLogEventFunction	445
13.13.1.23 spinLogEventHandler	445
13.13.1.24 spinRemovalEventFunction	445
13.13.1.25 spinSystem	445
13.13.1.26 spinVideo	445
13.13.2 Enumeration Type Documentation	445
13.13.2.1 _spinActionCommandStatus	445
13.13.2.2 _spinColorProcessingAlgorithm	446
13.13.2.3 _spinError	446

13.13.2.4 _spinImageFileFormat	448
13.13.2.5 _spinImageStatus	448
13.13.2.6 _spinLogLevel	449
13.13.2.7 _spinStatisticsChannel	449
13.13.2.8 _spinTIFFCompressionMethod	450
13.13.2.9 _spinTLPayloadType	450
13.13.2.10 _spinTLPixelFormatNamespace	451
13.13.3 Variable Documentation	451
13.13.3.1 False	451
13.13.3.2 True	451
13.14 include/spinc/SpinnakerGenApiC.h File Reference	452
13.14.1 Function Documentation	456
13.14.1.1 spinBooleanGetValue()	456
13.14.1.2 spinBooleanSetValue()	456
13.14.1.3 spinCategoryGetFeatureByIndex()	457
13.14.1.4 spinCategoryGetNumFeatures()	457
13.14.1.5 spinCategoryReleaseNode()	458
13.14.1.6 spinCommandExecute()	458
13.14.1.7 spinCommandIsDone()	459
13.14.1.8 spinEnumerationEntryGetEnumValue()	459
13.14.1.9 spinEnumerationEntryGetIntValue()	460
13.14.1.10 spinEnumerationEntryGetSymbolic()	460
13.14.1.11 spinEnumerationGetCurrentEntry()	461
13.14.1.12 spinEnumerationGetEntryByIndex()	461
13.14.1.13 spinEnumerationGetEntryByName()	462
13.14.1.14 spinEnumerationGetNumEntries()	462
13.14.1.15 spinEnumerationReleaseNode()	463
13.14.1.16 spinEnumerationSetEnumValue()	463
13.14.1.17 spinEnumerationSetIntValue()	464
13.14.1.18 spinFloatGetMax()	464
13.14.1.19 spinFloatGetMin()	465
13.14.1.20 spinFloatGetRepresentation()	465
13.14.1.21 spinFloatGetUnit()	466
13.14.1.22 spinFloatGetValue()	466
13.14.1.23 spinFloatGetValueEx()	467
13.14.1.24 spinFloatSetValue()	467
13.14.1.25 spinFloatSetValueEx()	468
13.14.1.26 spinIntegerGetInc()	468
13.14.1.27 spinIntegerGetMax()	469
13.14.1.28 spinIntegerGetMin()	469
13.14.1.29 spinIntegerGetRepresentation()	470
13.14.1.30 spinIntegerGetValue()	470

13.14.1.31 spinIntegerGetValueEx()	471
13.14.1.32 spinIntegerSetValue()	471
13.14.1.33 spinIntegerSetValueEx()	472
13.14.1.34 spinNodeDeregisterCallback()	472
13.14.1.35 spinNodeFromString()	473
13.14.1.36 spinNodeFromStringEx()	473
13.14.1.37 spinNodeGetAccessMode()	474
13.14.1.38 spinNodeGetCachingMode()	474
13.14.1.39 spinNodeGetDescription()	475
13.14.1.40 spinNodeGetDisplayName()	475
13.14.1.41 spinNodeGetImposedAccessMode()	476
13.14.1.42 spinNodeGetImposedVisibility()	476
13.14.1.43 spinNodeGetName()	477
13.14.1.44 spinNodeGetNameSpace()	477
13.14.1.45 spinNodeGetPollingTime()	478
13.14.1.46 spinNodeGetToolTip()	478
13.14.1.47 spinNodeGetType()	479
13.14.1.48 spinNodeGetVisibility()	479
13.14.1.49 spinNodeInvalidateNode()	480
13.14.1.50 spinNodeIsAvailable()	480
13.14.1.51 spinNodeIsEqual()	481
13.14.1.52 spinNodeIsImplemented()	481
13.14.1.53 spinNodeIsReadable()	482
13.14.1.54 spinNodeIsWritable()	482
13.14.1.55 spinNodeMapGetNode()	483
13.14.1.56 spinNodeMapGetNodeByIndex()	483
13.14.1.57 spinNodeMapGetNumNodes()	484
13.14.1.58 spinNodeMapPoll()	484
13.14.1.59 spinNodeMapReleaseNode()	485
13.14.1.60 spinNodeRegisterCallback()	485
13.14.1.61 spinNodeToString()	486
13.14.1.62 spinNodeToStringEx()	486
13.14.1.63 spinRegisterGet()	487
13.14.1.64 spinRegisterGetAddress()	487
13.14.1.65 spinRegisterGetEx()	488
13.14.1.66 spinRegisterGetLength()	488
13.14.1.67 spinRegisterSet()	489
13.14.1.68 spinRegisterSetEx()	489
13.14.1.69 spinRegisterSetReference()	490
13.14.1.70 spinStringGetMaxLength()	490
13.14.1.71 spinStringGetValue()	491
13.14.1.72 spinStringGetValueEx()	491

13.14.1.73 spinStringSetValue()	492
13.14.1.74 spinStringSetValueEx()	492
13.15 include/spinc/SpinnakerGenApiDefsC.h File Reference	493
13.15.1 Typedef Documentation	496
13.15.1.1 spinNodeCallbackFunction	496
13.15.1.2 spinNodeCallbackHandle	496
13.15.1.3 spinNodeHandle	496
13.15.1.4 spinNodeMapHandle	496
13.15.2 Enumeration Type Documentation	496
13.15.2.1 _spinAccessMode	496
13.15.2.2 _spinCachingMode	497
13.15.2.3 _spinDisplayNotation	497
13.15.2.4 _spinEndianess	498
13.15.2.5 _spinIncMode	498
13.15.2.6 _spinInputDirection	498
13.15.2.7 _spinInterfaceType	499
13.15.2.8 _spinLinkType	499
13.15.2.9 _spinNameSpace	500
13.15.2.10 _spinNodeType	500
13.15.2.11 _spinRepresentation	501
13.15.2.12 _spinSign	501
13.15.2.13 _spinSlope	501
13.15.2.14 _spinStandardNameSpace	502
13.15.2.15 _spinVisibility	502
13.15.2.16 _spinXMLValidation	503
13.15.2.17 _spinYesNo	503
13.16 include/spinc/SpinnakerPlatformC.h File Reference	504
13.16.1 Macro Definition Documentation	504
13.16.1.1 SPINNAKERC_API	504
13.17 include/spinc/SpinVideoC.h File Reference	505
13.17.1 Function Documentation	505
13.17.1.1 spinVideoAppend()	505
13.17.1.2 spinVideoClose()	506
13.17.1.3 spinVideoOpenH264()	506
13.17.1.4 spinVideoOpenMJPEG()	506
13.17.1.5 spinVideoOpenUncompressed()	506
13.17.1.6 spinVideoSetMaximumFileSize()	506
13.18 include/spinc/TransportLayerDefsC.h File Reference	507
13.18.1 Enumeration Type Documentation	509
13.18.1.1 _spinTLDeviceAccessStatusEnums	509
13.18.1.2 _spinTLDeviceCurrentSpeedEnums	509
13.18.1.3 _spinTLDeviceEndianessMechanismEnums	509

13.18.1.4 _spinTLDeviceTypeEnums	511
13.18.1.5 _spinTLFilterDriverStatusEnums	511
13.18.1.6 _spinTLGenICamXMLLocationEnums	511
13.18.1.7 _spinTLGevCCPEnums	512
13.18.1.8 _spinTLGUIXMLLocationEnums	512
13.18.1.9 _spinTLInterfaceTypeEnums	512
13.18.1.10 _spinTLPOEStatusEnums	513
13.18.1.11 _spinTLStreamBufferCountModeEnums	513
13.18.1.12 _spinTLStreamBufferHandlingModeEnums	513
13.18.1.13 _spinTLStreamModeEnums	514
13.18.1.14 _spinTLStreamTypeEnums	514
13.18.1.15 _spinTLTLTypeEnums	515
13.19 include/spinc/TransportLayerDeviceC.h File Reference	515
13.20 include/spinc/TransportLayerInterfaceC.h File Reference	516
13.21 include/spinc/TransportLayerStreamC.h File Reference	517
13.22 include/spinc/TransportLayerSystemC.h File Reference	517

Index

519

Chapter 1

Getting Started

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

- [Benefits of Spinnaker](#)
- [Software Licensing Information](#)
- [Software Maintenance Policy](#)
- [FlyCapture2 Feature Comparison with Spinnaker](#)
- [Programmer's Guide](#)
- [Working with GenICam GenTL Devices](#)
- [Myricom](#)

Chapter 2

Programmer's Guide

Chapter 3

Benefits of Spinnaker

Please see (<http://softwareservices.flir.com/Spinnaker/latest/index.html>) for the latest version of this document

Chapter 4

FlyCapture2 Feature Comparison with Spinnaker

Please see (<http://softwareservices.flir.com/Spinnaker/latest/page3.html>) for the latest version of this document

Chapter 5

Working with GenICam GenTL Devices

5.1 GenTL Overview

FLIR GenTL Producer is a software driver that implements the GenICam™ GenTL 1.5 standard (<https://www.emva.org/>). It allows users to enumerate, communicate and stream from FLIR GigE Vision and USB3 Vision devices in a generic way independent from the underlying transport technology. This allows third-party software such as MATLAB (<https://www.mathworks.com>) and other software libraries to work with FLIR devices in a transport layer agnostic way. These applications are referred to as "GenTL Consumers," which directly use one or more GenTL Producers.

NOTE: Consumer applications must be aware of differences in device capabilities and be prepared to handle specific device models differently.

5.2 Installation

In order to use a FLIR GenTL producer, it needs to be properly registered and installed on the system. **The FLIR Producer comes packaged with the full Spinnaker SDK installer as of 2.x or newer.**

The GenTL Producer is provided as a platform dependent, dynamic loadable library file with the `.cti` ("Common Transport Interface") extension.

The Spinnaker SDK installer stores the folder paths for 32-bit and 64-bit GenTL Producers (`.cti` files) in environment variables named `GENICAM_GENTL32_PATH` and `GENICAM_GENTL64_PATH`, respectively. If there are multiple GenTL Producers installed on the system, path entries must be separated by `;` on Windows and `:` on UNIX-like systems.

NOTE: A 32bit GenTL consumer application will require a 32-bit GenTL producer and a 64-bit application will require a 64-bit producer library.

5.3 Troubleshooting

5.3.1 Enable FLIR GenTL Logging

FLIR GenTL Logging can be enabled if a configuration file with the name "log4cpp.gentl.property" resides in the path of where the consumer application executes from. For MATLAB, this is where the working directory is set and may default to the "Downloads" folder on Windows.

Sample log4cpp.gentl.property configuration file:

```
# FLIR GenTL Property Configuration file
log4cpp.rootCategory=ERROR, rootAppender
log4cpp.category.GenTLCategory=ERROR, GenTLCategory

log4cpp.appender.rootAppender=ConsoleAppender
log4cpp.appender.rootAppender.layout=PatternLayout
log4cpp.appender.rootAppender.layout.ConversionPattern=[%p] %d [%t] %m%n

log4cpp.appender.GenTLCategory=RollingFileAppender
log4cpp.appender.GenTLCategory.fileName=$(ALLUSERSPROFILE)\Spinnaker\Logs\GenTL.log
log4cpp.appender.GenTLCategory.append=true
log4cpp.appender.GenTLCategory.maxFileSize=1000000
log4cpp.appender.GenTLCategory.maxBackupIndex=5
log4cpp.appender.GenTLCategory.layout=PatternLayout
log4cpp.appender.GenTLCategory.layout.ConversionPattern=[%p] %d [%t] %m%n
```

5.3.2 USB3 Device Image Tearing

Image tearing could occur with certain USB3 host controllers when streaming with a GenTL producer. To work around the issue, make sure the size of each buffer announced to the FLIR GenTL producer is 1024 bytes aligned. The size of each buffer should be $(\text{bufferSize} + 1024 - 1) / 1024 * 1024$ where 1024 is the USB3 packet transfer size.

For more information about image tearing causes and solutions, please refer to: <https://www.flir.com/support-center/iis/machine-vision/application-note/image-tearing-causes-and-solutions/>

Chapter 6

Software Licensing Information

Table 6.1 License table

Component	License
Spinnaker	Copyright (c) 2001-2020 FLIR Systems, 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	GenICam License
AdapterList	The Code Project Open License (CPO-OL)
Make ListView.ScrollIntoView Scroll the Item into the Center of the ListView	WP:CC-BY-SA License
Work with Bitmaps Faster in C#	The Code Project Open License (CPO-OL) 1.02
FreeImage	FreeImage public license
Boost	Boost Software License
Libusb	LGPLv2.1 License
Libraw1394	LGPLv2.0 License
FFMPEG	LGPLv2.1 License
log4Net	Apache license 2.0
log4Cpp	LGPL License

The licenses mentioned above can also be found in the Spinnaker installed license folder.

Chapter 7

Software Maintenance Policy

7.1 GenTL Overview

This document outlines the FLIR maintenance policy for Spinnaker Software Development Kit (SDK). FLIR regularly provides SDK updates that may contain support for new or updated features, enhancements, updated drivers, updated examples, bug fixes or documentation updates. Updates may also address changes with introducing and/or deprecating language runtimes, operating systems and dependencies.

We recommend users to stay up-to-date with SDK releases to keep up with the latest features, bug fixes and performance improvements. Continued use of an unsupported SDK version is not recommended and is done at the user's discretion.

Spinnaker SDK releases are published through our website and can be found here: <https://www.flir.ca/products/spinnaker-sdk/>

7.2 Platform Support Policy

7.2.1 Windows Support

FLIR will continue to maintain, fix and build the last two major versions of Spinnaker SDK against the latest available version of Windows x86/x64. The latest three versions of Visual Studio compiler toolchain are supported on Windows. Only the latest compiler toolchain on the latest available version of Windows are being actively tested.

7.2.2 Linux Desktop Support

FLIR will continue to maintain, fix and build the last two major versions of Spinnaker SDK against the latest two LTS versions of Ubuntu x86/x64. Only the latest x64 LTS version of Ubuntu is being actively tested.

7.2.3 Linux Embedded Support

FLIR will continue to maintain, fix and build the last two major versions of Spinnaker SDK against the latest supported LTS version of Ubuntu ARMHF/ARM64 for a specific board. Only the latest LTS Ubuntu version on an ARM64 board is being actively tested. Contact sales if you need support for a specific embedded board.

7.2.4 MacOS Support

FLIR will continue to maintain, fix and build the last two major versions of Spinnaker SDK against MacOS Mojave (10.14). Contact sales if you need newer MacOS support.

7.3 Versioning Policy

Spinnaker SDK releases use a modified semantic versioning scheme and is indicated by four sets of numbers separated by periods:

MAJOR.MINOR.0.PATCH

- MAJOR: Version change that can include incompatible API changes
- MINOR: Version change that adds functionality in a backwards-compatible manner
- PATCH: Version change with backwards-compatible fixes

Reference: <https://www.flir.com/support-center/iis/machine-vision/knowledge-base/flir-mac>

Chapter 8

Module Index

8.1 Modules

Here is a list of all modules:

Camera Enumerations	22
Chunk Data Structures	23
Spinnaker C QuickSpin API	24
TLDevice Structures	64
TLInterface Structures	65
TLStream Structures	66
TlSystem Structures	67
QuickSpin Access	25
Spinnaker C API	26
Spinnaker C Definitions	21
Error Handling	27
System Access	28
InterfaceList Access	29
CameraList Access	30
ImageList Access	31
Interface Access	32
Camera Access	33
Image Access	34
Image Processor Access	35
Event Access	38
ImageStatistics Access	39
Logging Event Data Access	40
Device Event Data Access	41
Chunk data access	42
Spinnaker C Handles	43
Spinnaker C Function Signatures	44
Spinnaker C Enumerations	45
Spinnaker C Structures	46
Spinnaker C GenICam API	47
Node Map Access	48
Node Access	49
IValue Access	50
String Access	51
Integer Access	52
IFloat Access	53

IEnumeration Access	54
IEnumEntry Access	55
IBoolean Access	56
ICommand Access	57
ICategory Access	58
IRegister Access	59
Spinnaker C GenICam Handles	60
Spinnaker C GenICam Enumerations	61
SpinVideo Recording Access	62
Transport Layer Enumerations	63

Chapter 9

Data Structure Index

9.1 Data Structures

Here are the data structures with brief descriptions:

_actionCommandResult	
Action Command Result	69
_quickSpin	70
_quickSpinTLDevice	167
_quickSpinTLInterface	173
_quickSpinTLStream	181
_quickSpinTLSystem	186
_spinAVIOption	
Options for saving uncompressed videos	190
_spinBMPOption	
Options for saving BMP images	191
_spinChunkData	
The type of information that can be obtained from image chunk data	192
_spinH264Option	
Options for saving H264 videos	198
_spinJPEGOption	
Options for saving JPEG images	199
_spinJPG2Option	
Options for saving JPEG 2000 images	200
_spinLibraryVersion	
Provides easier access to the current version of Spinnaker	201
_spinMJPGOption	
Options for saving MJPG videos	202
_spinPGMOption	
Options for saving PGM images	204
_spinPNGOption	
Options for saving PNG images	205
_spinPPMOption	
Options for saving PPM images	206
_spinTIFFOption	
Options for saving TIFF images	207

Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

include/spinc/ CameraDefsC.h	209
include/spinc/ ChunkDataDefC.h	326
include/spinc/ QuickSpinC.h	327
include/spinc/ QuickSpinDefsC.h	329
include/spinc/ SpinnakerC.h	331
include/spinc/ SpinnakerDefsC.h	436
include/spinc/ SpinnakerGenApiC.h	452
include/spinc/ SpinnakerGenApiDefsC.h	493
include/spinc/ SpinnakerPlatformC.h	504
include/spinc/ SpinVideoC.h	505
include/spinc/ TransportLayerDefsC.h	507
include/spinc/ TransportLayerDeviceC.h	515
include/spinc/ TransportLayerInterfaceC.h	516
include/spinc/ TransportLayerStreamC.h	517
include/spinc/ TransportLayerSystemC.h	517

Chapter 11

Module Documentation

11.1 Spinnaker C Definitions

Collaboration diagram for Spinnaker C Definitions:



Definitions for Spinnaker C

Definitions for Spinnaker C API

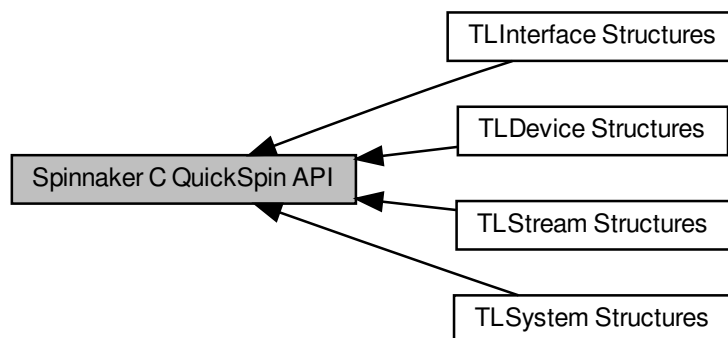
Holds enumerations, typedefs and structures that are used across the Spinnaker C API wrapper.

11.2 Camera Enumerations

11.3 Chunk Data Structures

11.4 Spinnaker C QuickSpin API

Collaboration diagram for Spinnaker C QuickSpin API:



Modules

- [TLDevice Structures](#)
- [TLInterface Structures](#)
- [TLStream Structures](#)
- [TLSystem Structures](#)

11.4.1 Detailed Description

11.5 QuickSpin Access

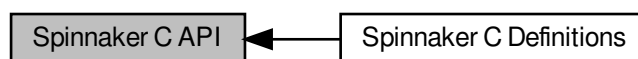
The functions in this section initialize the various QuickSpin structs for the C API.

The functions in this section initialize the various QuickSpin structs for the C API.

11.6 Spinnaker C API

SpinnakerPlatform C Include.

Collaboration diagram for Spinnaker C API:



Modules

- [Spinnaker C Definitions](#)

11.6.1 Detailed Description

SpinnakerPlatform C Include.

Spinnaker C Definition Includes Spinnaker GenICam C Wrapper Includes Spinnaker QuickSpin C Includes

Spinnaker C Definition Includes

11.7 Error Handling

The functions in this section provide access to additional information related to error returns.

The functions in this section provide access to additional information related to error returns.

11.8 System Access

The functions in this section provide access to information, objects, and functionality of the system object. This includes the system object, interface and camera lists, and interface and logging events.

The functions in this section provide access to information, objects, and functionality of the system object. This includes the system object, interface and camera lists, and interface and logging events.

11.9 InterfaceList Access

The functions in this section provide access to information, objects, and functionality of interface lists. This includes updating, size and interface retrieval, and clearance.

The functions in this section provide access to information, objects, and functionality of interface lists. This includes updating, size and interface retrieval, and clearance.

11.10 CameraList Access

The functions in this section provide access to information, objects, and functionality of camera lists. This includes updating, size and camera retrieval, and clearance.

The functions in this section provide access to information, objects, and functionality of camera lists. This includes updating, size and camera retrieval, and clearance.

11.11 ImageList Access

The functions in this section provide access to information, objects, and functionality of image lists. This includes updating, size and image retrieval, and clearance.

The functions in this section provide access to information, objects, and functionality of image lists. This includes updating, size and image retrieval, and clearance.

11.12 Interface Access

The functions in this section provide access to information, objects, and functionality of interfaces. This includes camera list and nodemap retrieval, event handler registration, and interface release.

The functions in this section provide access to information, objects, and functionality of interfaces. This includes camera list and nodemap retrieval, event handler registration, and interface release.

11.13 Camera Access

The functions in this section provide access to information, objects, and functionality of cameras. This includes nodemap retrieval, acquisition and init commands, event handler registration, and camera property retrieval.

The functions in this section provide access to information, objects, and functionality of cameras. This includes nodemap retrieval, acquisition and init commands, event handler registration, and camera property retrieval.

11.14 Image Access

The functions in this section provide access to information and functionality of images. This includes creation, destruction, and saving as well as a wealth of information including things like width, height, stride, and timestamp.

The functions in this section provide access to information and functionality of images. This includes creation, destruction, and saving as well as a wealth of information including things like width, height, stride, and timestamp.

11.15 Image Processor Access

The functions in this section provide access to information and functionality of image processor. This includes image processor creation, deletion, image conversion, image decompression and image post processing methods.

The functions in this section provide access to information and functionality of image processor. This includes image processor creation, deletion, image conversion, image decompression and image post processing methods.

All supported input image pixel formats can be converted to supported output image pixel formats. If the input pixel format is a compressed format, the decompression will occur before converting to the output pixel format.

List of supported input image pixel formats:

- PixelFormat_Mono8
- PixelFormat_Mono16
- 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_Mono10
- PixelFormat_Mono12
- PixelFormat_Mono14
- PixelFormat_BayerBG10
- PixelFormat_BayerBG12
- PixelFormat_BayerGB10
- PixelFormat_BayerGB12
- PixelFormat_BayerGR10
- PixelFormat_BayerGR12
- PixelFormat_BayerRG10
- PixelFormat_BayerRG12
- PixelFormat_RGBa8
- PixelFormat_RGB8
- PixelFormat_BGR16
- PixelFormat_R12
- PixelFormat_G12
- PixelFormat_B12
- PixelFormat_YUV8_UYV
- PixelFormat_YUV411_8_UYYVYY
- PixelFormat_YUV422_8
- PixelFormat_Polarized8
- PixelFormat_Polarized10p
- PixelFormat_Polarized12p
- PixelFormat_Polarized16
- PixelFormat_BayerRGPolarized8

- PixelFormat_BayerRGPolarized10p
- PixelFormat_BayerRGPolarized12p
- PixelFormat_BayerRGPolarized16
- PixelFormat_LLCMono8
- PixelFormat_LLCBayerRG8
- PixelFormat_JPEGMono8
- PixelFormat_JPEGColor8
- PixelFormat_Raw16
- PixelFormat_Raw8
- PixelFormat_R12_Jpeg
- PixelFormat_GR12_Jpeg
- PixelFormat_GB12_Jpeg
- PixelFormat_B12_Jpeg

List of supported output image pixel formats

- PixelFormat_Mono8
- PixelFormat_Mono16
- PixelFormat_BayerBG8
- PixelFormat_BayerGB8
- PixelFormat_BayerRG8
- PixelFormat_BayerGR8
- PixelFormat_BayerBG16
- PixelFormat_BayerGB16
- PixelFormat_BayerRG16
- PixelFormat_BayerGR16
- PixelFormat_BGR8
- PixelFormat_BGRa8
- PixelFormat_RGB8
- PixelFormat_RGBa8
- PixelFormat_BGR16
- PixelFormat_RGB16
- PixelFormat_R12
- PixelFormat_G12
- PixelFormat_B12

11.16 Event Access

The functions in this section allow for the creation and destruction of events.

The functions in this section allow for the creation and destruction of events.

11.17 ImageStatistics Access

The functions in this section provide access to information and functionality related to image statistics. This includes context creation and destruction, the enabling and disabling of channels, and value retrieval.

The functions in this section provide access to information and functionality related to image statistics. This includes context creation and destruction, the enabling and disabling of channels, and value retrieval.

11.18 Logging Event Data Access

The functions in this section allow for the retrieval of logging event data.

The functions in this section allow for the retrieval of logging event data.

11.19 Device Event Data Access

The functions in this section allow for the retrieval of device event data.

The functions in this section allow for the retrieval of device event data.

11.20 Chunk data access

The functions in this section provide access to chunk data stored on images.

The functions in this section provide access to chunk data stored on images.

11.21 Spinnaker C Handles

Spinnaker C handle definitions.

Spinnaker C handle definitions.

11.22 Spinnaker C Function Signatures

Spinnaker C function signature definitions.

Spinnaker C function signature definitions.

11.23 Spinnaker C Enumerations

Spinnaker C enumeration definitions.

Spinnaker C enumeration definitions.

11.24 Spinnaker C Structures

Spinnaker C structure definitions.

Spinnaker C structure definitions.

11.25 Spinnaker C GenICam API

11.26 Node Map Access

The functions in this section provide access to information, objects, and functionality related to nodemaps. This includes nodes, node counts, and polling.

The functions in this section provide access to information, objects, and functionality related to nodemaps. This includes nodes, node counts, and polling.

11.27 Node Access

The functions in this section provide access to information and objects retrieved from nodes. This includes node properties and callback registration.

The functions in this section provide access to information and objects retrieved from nodes. This includes node properties and callback registration.

11.28 IValue Access

The functions in this section provide access to nodes as value nodes. As value nodes are not an actual node type, the functions are named as regular nodes. Functions include reading from and writing to any node with a string.

The functions in this section provide access to nodes as value nodes. As value nodes are not an actual node type, the functions are named as regular nodes. Functions include reading from and writing to any node with a string.

11.29 String Access

The functions in this section provide access to string nodes using character pointers and arrays. This includes getters and setters of values and value lengths.

The functions in this section provide access to string nodes using character pointers and arrays. This includes getters and setters of values and value lengths.

11.30 Integer Access

The functions in this section provide access to integer nodes using the `int64_t` data type. This includes value getters and setters, min, max, and increment functions, and node representation.

The functions in this section provide access to integer nodes using the `int64_t` data type. This includes value getters and setters, min, max, and increment functions, and node representation.

11.31 IFloat Access

The functions in this section provide access to float nodes using double as the data type. This includes value getters and setters, min and max functions, and node representation.

The functions in this section provide access to float nodes using double as the data type. This includes value getters and setters, min and max functions, and node representation.

11.32 IEnumeration Access

The functions in this section provide access to enum nodes. This includes retrieving the number of entries, an entry by index or name, retrieving the current entry node, or setting the node using an integer.

The functions in this section provide access to enum nodes. This includes retrieving the number of entries, an entry by index or name, retrieving the current entry node, or setting the node using an integer.

11.33 IEnumEntry Access

The functions in this section provide access to entry nodes. This includes retrieving the integer value or the symbolic of an entry.

The functions in this section provide access to entry nodes. This includes retrieving the integer value or the symbolic of an entry.

11.34 IBoolean Access

The functions in this section provide access to boolean nodes using the `bool8_t` data type, values represented with 'True' and 'False'. This includes value getters and setters.

The functions in this section provide access to boolean nodes using the `bool8_t` data type, values represented with 'True' and 'False'. This includes value getters and setters.

11.35 ICommand Access

The functions in this section all provide access to information and objects retrieved from nodes. This includes node properties and callbacks.

The functions in this section all provide access to information and objects retrieved from nodes. This includes node properties and callbacks.

11.36 ICategory Access

The functions in this section all provide access to information and objects retrieved from nodes. This includes node properties and callbacks.

The functions in this section all provide access to information and objects retrieved from nodes. This includes node properties and callbacks.

11.37 IRegister Access

The functions in this section provide access to register nodes. This includes access to the node, its address and length, and reference.

The functions in this section provide access to register nodes. This includes access to the node, its address and length, and reference.

11.38 Spinnaker C GenICam Handles

Handle definitions for Spinnaker C GenICam API.

Handle definitions for Spinnaker C GenICam API.

11.39 Spinnaker C GenICam Enumerations

Enumeration definitions for Spinnaker C GenICam API.

Enumeration definitions for Spinnaker C GenICam API.

11.40 SpinVideo Recording Access

The functions in this section provide access to video recording capabilities, which include opening, building, and closing video files.

The functions in this section provide access to video recording capabilities, which include opening, building, and closing video files.

11.41 Transport Layer Enumerations

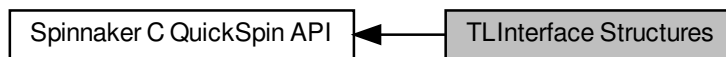
11.42 TLDevice Structures

Collaboration diagram for TLDevice Structures:



11.43 TLInterface Structures

Collaboration diagram for TLInterface Structures:



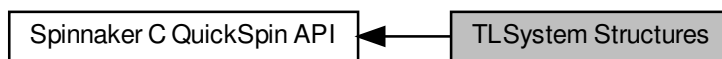
11.44 TLStream Structures

Collaboration diagram for TLStream Structures:



11.45 TLSystem Structures

Collaboration diagram for TLSystem Structures:



Chapter 12

Data Structure Documentation

12.1 `_actionCommandResult` Struct Reference

Action Command Result.

Data Fields

- unsigned int [DeviceAddress](#)
- spinActionCommandStatus [Status](#)

12.1.1 Detailed Description

Action Command Result.

12.1.2 Field Documentation

12.1.2.1 `DeviceAddress`

```
unsigned int DeviceAddress
```

12.1.2.2 `Status`

```
spinActionCommandStatus Status
```

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

- include/spinc/[SpinnakerDefsC.h](#)

12.2 `_quickSpin` Struct Reference

Data Fields

- `quickSpinIntegerNode` LUTIndex
- `quickSpinBooleanNode` LUTEnable
- `quickSpinIntegerNode` LUTValue
- `quickSpinEnumerationNode` LUTSelector
- `quickSpinFloatNode` ExposureTime
- `quickSpinCommandNode` AcquisitionStop
- `quickSpinFloatNode` AcquisitionResultingFrameRate
- `quickSpinFloatNode` AcquisitionLineRate
- `quickSpinCommandNode` AcquisitionStart
- `quickSpinCommandNode` TriggerSoftware
- `quickSpinEnumerationNode` ExposureMode
- `quickSpinEnumerationNode` AcquisitionMode
- `quickSpinIntegerNode` AcquisitionFrameCount
- `quickSpinEnumerationNode` TriggerSource
- `quickSpinEnumerationNode` TriggerActivation
- `quickSpinEnumerationNode` SensorShutterMode
- `quickSpinFloatNode` TriggerDelay
- `quickSpinEnumerationNode` TriggerMode
- `quickSpinFloatNode` AcquisitionFrameRate
- `quickSpinEnumerationNode` TriggerOverlap
- `quickSpinEnumerationNode` TriggerSelector
- `quickSpinBooleanNode` AcquisitionFrameRateEnable
- `quickSpinEnumerationNode` ExposureAuto
- `quickSpinIntegerNode` AcquisitionBurstFrameCount
- `quickSpinIntegerNode` EventTest
- `quickSpinIntegerNode` EventTestTimestamp
- `quickSpinIntegerNode` EventExposureEndFrameID
- `quickSpinIntegerNode` EventExposureEnd
- `quickSpinIntegerNode` EventExposureEndTimestamp
- `quickSpinIntegerNode` EventError
- `quickSpinIntegerNode` EventErrorTimestamp
- `quickSpinIntegerNode` EventErrorCode
- `quickSpinIntegerNode` EventErrorFrameID
- `quickSpinEnumerationNode` EventSelector
- `quickSpinBooleanNode` EventSerialReceiveOverflow
- `quickSpinIntegerNode` EventSerialPortReceive
- `quickSpinIntegerNode` EventSerialPortReceiveTimestamp
- `quickSpinStringNode` EventSerialData
- `quickSpinIntegerNode` EventSerialDataLength
- `quickSpinEnumerationNode` EventNotification
- `quickSpinIntegerNode` LogicBlockLUTRowIndex
- `quickSpinEnumerationNode` LogicBlockSelector
- `quickSpinEnumerationNode` LogicBlockLUTInputActivation
- `quickSpinEnumerationNode` LogicBlockLUTInputSelector
- `quickSpinEnumerationNode` LogicBlockLUTInputSource
- `quickSpinBooleanNode` LogicBlockLUTOutputValue
- `quickSpinIntegerNode` LogicBlockLUTOutputValueAll
- `quickSpinEnumerationNode` LogicBlockLUTSelector
- `quickSpinFloatNode` ColorTransformationValue
- `quickSpinBooleanNode` ColorTransformationEnable

- [quickSpinEnumerationNode ColorTransformationSelector](#)
- [quickSpinEnumerationNode RgbTransformLightSource](#)
- [quickSpinFloatNode Saturation](#)
- [quickSpinBooleanNode SaturationEnable](#)
- [quickSpinEnumerationNode ColorTransformationValueSelector](#)
- [quickSpinIntegerNode TimestampLatchValue](#)
- [quickSpinCommandNode TimestampReset](#)
- [quickSpinStringNode DeviceUserID](#)
- [quickSpinFloatNode DeviceTemperature](#)
- [quickSpinIntegerNode MaxDeviceResetTime](#)
- [quickSpinIntegerNode DeviceTLVersionMinor](#)
- [quickSpinStringNode DeviceSerialNumber](#)
- [quickSpinStringNode DeviceVendorName](#)
- [quickSpinEnumerationNode DeviceRegistersEndianness](#)
- [quickSpinStringNode DeviceManufacturerInfo](#)
- [quickSpinIntegerNode DeviceLinkSpeed](#)
- [quickSpinIntegerNode LinkUptime](#)
- [quickSpinIntegerNode DeviceEventChannelCount](#)
- [quickSpinCommandNode TimestampLatch](#)
- [quickSpinEnumerationNode DeviceScanType](#)
- [quickSpinCommandNode DeviceReset](#)
- [quickSpinEnumerationNode DeviceCharacterSet](#)
- [quickSpinIntegerNode DeviceLinkThroughputLimit](#)
- [quickSpinStringNode DeviceFirmwareVersion](#)
- [quickSpinIntegerNode DeviceStreamChannelCount](#)
- [quickSpinEnumerationNode DeviceTLType](#)
- [quickSpinStringNode DeviceVersion](#)
- [quickSpinEnumerationNode DevicePowerSupplySelector](#)
- [quickSpinStringNode SensorDescription](#)
- [quickSpinStringNode DeviceModelName](#)
- [quickSpinIntegerNode DeviceTLVersionMajor](#)
- [quickSpinEnumerationNode DeviceTemperatureSelector](#)
- [quickSpinIntegerNode EnumerationCount](#)
- [quickSpinFloatNode PowerSupplyCurrent](#)
- [quickSpinStringNode DeviceID](#)
- [quickSpinIntegerNode DeviceUptime](#)
- [quickSpinIntegerNode DeviceLinkCurrentThroughput](#)
- [quickSpinIntegerNode DeviceMaxThroughput](#)
- [quickSpinCommandNode FactoryReset](#)
- [quickSpinFloatNode PowerSupplyVoltage](#)
- [quickSpinEnumerationNode DeviceIndicatorMode](#)
- [quickSpinFloatNode DeviceLinkBandwidthReserve](#)
- [quickSpinIntegerNode AasRoiOffsetY](#)
- [quickSpinIntegerNode AasRoiOffsetX](#)
- [quickSpinEnumerationNode AutoExposureControlPriority](#)
- [quickSpinFloatNode BalanceWhiteAutoLowerLimit](#)
- [quickSpinFloatNode BalanceWhiteAutoDamping](#)
- [quickSpinIntegerNode AasRoiHeight](#)
- [quickSpinFloatNode AutoExposureGreyValueUpperLimit](#)
- [quickSpinFloatNode AutoExposureTargetGreyValue](#)
- [quickSpinFloatNode AutoExposureGainLowerLimit](#)
- [quickSpinFloatNode AutoExposureGreyValueLowerLimit](#)
- [quickSpinEnumerationNode AutoExposureMeteringMode](#)
- [quickSpinFloatNode AutoExposureExposureTimeUpperLimit](#)
- [quickSpinFloatNode AutoExposureGainUpperLimit](#)

- [quickSpinFloatNode AutoExposureControlLoopDamping](#)
- [quickSpinFloatNode AutoExposureEVCompensation](#)
- [quickSpinFloatNode AutoExposureExposureTimeLowerLimit](#)
- [quickSpinEnumerationNode BalanceWhiteAutoProfile](#)
- [quickSpinEnumerationNode AutoAlgorithmSelector](#)
- [quickSpinEnumerationNode AutoExposureTargetGreyValueAuto](#)
- [quickSpinBooleanNode AasRoiEnable](#)
- [quickSpinEnumerationNode AutoExposureLightingMode](#)
- [quickSpinIntegerNode AasRoiWidth](#)
- [quickSpinFloatNode BalanceWhiteAutoUpperLimit](#)
- [quickSpinIntegerNode LinkErrorCount](#)
- [quickSpinBooleanNode GevCurrentIPConfigurationDHCP](#)
- [quickSpinIntegerNode GevInterfaceSelector](#)
- [quickSpinIntegerNode GevSCPD](#)
- [quickSpinIntegerNode GevTimestampTickFrequency](#)
- [quickSpinIntegerNode GevSCPSPacketSize](#)
- [quickSpinIntegerNode GevCurrentDefaultGateway](#)
- [quickSpinBooleanNode GevSCCFGUnconditionalStreaming](#)
- [quickSpinIntegerNode GevMCTT](#)
- [quickSpinBooleanNode GevSCPSDoNotFragment](#)
- [quickSpinIntegerNode GevCurrentSubnetMask](#)
- [quickSpinIntegerNode GevStreamChannelSelector](#)
- [quickSpinIntegerNode GevCurrentIPAddress](#)
- [quickSpinIntegerNode GevMCSP](#)
- [quickSpinIntegerNode GevGVCPPendingTimeout](#)
- [quickSpinEnumerationNode GevIEEE1588Status](#)
- [quickSpinStringNode GevFirstURL](#)
- [quickSpinIntegerNode GevMACAddress](#)
- [quickSpinIntegerNode GevPersistentSubnetMask](#)
- [quickSpinIntegerNode GevMCPHostPort](#)
- [quickSpinIntegerNode GevSCPHostPort](#)
- [quickSpinBooleanNode GevGVCPPendingAck](#)
- [quickSpinIntegerNode GevSCPInterfaceIndex](#)
- [quickSpinBooleanNode GevSupportedOption](#)
- [quickSpinEnumerationNode GevIEEE1588Mode](#)
- [quickSpinBooleanNode GevCurrentIPConfigurationLLA](#)
- [quickSpinIntegerNode GevSCSP](#)
- [quickSpinBooleanNode GevIEEE1588](#)
- [quickSpinBooleanNode GevSCCFGExtendedChunkData](#)
- [quickSpinIntegerNode GevPersistentIPAddress](#)
- [quickSpinBooleanNode GevCurrentIPConfigurationPersistentIP](#)
- [quickSpinEnumerationNode GevIEEE1588ClockAccuracy](#)
- [quickSpinIntegerNode GevHeartbeatTimeout](#)
- [quickSpinIntegerNode GevPersistentDefaultGateway](#)
- [quickSpinEnumerationNode GevCCP](#)
- [quickSpinIntegerNode GevMCDA](#)
- [quickSpinIntegerNode GevSCDA](#)
- [quickSpinIntegerNode GevSCPDirection](#)
- [quickSpinBooleanNode GevSCPSFireTestPacket](#)
- [quickSpinStringNode GevSecondURL](#)
- [quickSpinEnumerationNode GevSupportedOptionSelector](#)
- [quickSpinBooleanNode GevGVCPHeartbeatDisable](#)
- [quickSpinIntegerNode GevMCRC](#)
- [quickSpinBooleanNode GevSCPSBigEndian](#)
- [quickSpinIntegerNode GevNumberOfInterfaces](#)

- quickSpinIntegerNode TLParamsLocked
- quickSpinIntegerNode PayloadSize
- quickSpinIntegerNode PacketResendRequestCount
- quickSpinBooleanNode SharpeningEnable
- quickSpinEnumerationNode BlackLevelSelector
- quickSpinBooleanNode GammaEnable
- quickSpinBooleanNode SharpeningAuto
- quickSpinBooleanNode BlackLevelClampingEnable
- quickSpinFloatNode BalanceRatio
- quickSpinEnumerationNode BalanceWhiteAuto
- quickSpinFloatNode SharpeningThreshold
- quickSpinEnumerationNode GainAuto
- quickSpinFloatNode Sharpening
- quickSpinFloatNode Gain
- quickSpinEnumerationNode BalanceRatioSelector
- quickSpinEnumerationNode GainSelector
- quickSpinFloatNode BlackLevel
- quickSpinIntegerNode BlackLevelRaw
- quickSpinFloatNode Gamma
- quickSpinIntegerNode DefectTableIndex
- quickSpinCommandNode DefectTableFactoryRestore
- quickSpinIntegerNode DefectTableCoordinateY
- quickSpinCommandNode DefectTableSave
- quickSpinEnumerationNode DefectCorrectionMode
- quickSpinIntegerNode DefectTableCoordinateX
- quickSpinIntegerNode DefectTablePixelCount
- quickSpinBooleanNode DefectCorrectStaticEnable
- quickSpinCommandNode DefectTableApply
- quickSpinBooleanNode UserSetFeatureEnable
- quickSpinCommandNode UserSetSave
- quickSpinEnumerationNode UserSetSelector
- quickSpinCommandNode UserSetLoad
- quickSpinEnumerationNode UserSetDefault
- quickSpinEnumerationNode SerialPortBaudRate
- quickSpinIntegerNode SerialPortDataBits
- quickSpinEnumerationNode SerialPortParity
- quickSpinIntegerNode SerialTransmitQueueMaxCharacterCount
- quickSpinIntegerNode SerialReceiveQueueCurrentCharacterCount
- quickSpinEnumerationNode SerialPortSelector
- quickSpinEnumerationNode SerialPortStopBits
- quickSpinCommandNode SerialReceiveQueueClear
- quickSpinIntegerNode SerialReceiveFramingErrorCount
- quickSpinIntegerNode SerialTransmitQueueCurrentCharacterCount
- quickSpinIntegerNode SerialReceiveParityErrorCount
- quickSpinEnumerationNode SerialPortSource
- quickSpinIntegerNode SerialReceiveQueueMaxCharacterCount
- quickSpinIntegerNode SequencerSetStart
- quickSpinEnumerationNode SequencerMode
- quickSpinEnumerationNode SequencerConfigurationValid
- quickSpinEnumerationNode SequencerSetValid
- quickSpinIntegerNode SequencerSetSelector
- quickSpinEnumerationNode SequencerTriggerActivation
- quickSpinEnumerationNode SequencerConfigurationMode
- quickSpinCommandNode SequencerSetSave
- quickSpinEnumerationNode SequencerTriggerSource

- [quickSpinIntegerNode SequencerSetActive](#)
- [quickSpinIntegerNode SequencerSetNext](#)
- [quickSpinCommandNode SequencerSetLoad](#)
- [quickSpinIntegerNode SequencerPathSelector](#)
- [quickSpinBooleanNode SequencerFeatureEnable](#)
- [quickSpinIntegerNode TransferBlockCount](#)
- [quickSpinCommandNode TransferStart](#)
- [quickSpinIntegerNode TransferQueueMaxBlockCount](#)
- [quickSpinIntegerNode TransferQueueCurrentBlockCount](#)
- [quickSpinEnumerationNode TransferQueueMode](#)
- [quickSpinEnumerationNode TransferOperationMode](#)
- [quickSpinCommandNode TransferStop](#)
- [quickSpinIntegerNode TransferQueueOverflowCount](#)
- [quickSpinEnumerationNode TransferControlMode](#)
- [quickSpinFloatNode ChunkBlackLevel](#)
- [quickSpinIntegerNode ChunkFrameID](#)
- [quickSpinStringNode ChunkSerialData](#)
- [quickSpinFloatNode ChunkExposureTime](#)
- [quickSpinIntegerNode ChunkCompressionMode](#)
- [quickSpinFloatNode ChunkCompressionRatio](#)
- [quickSpinBooleanNode ChunkSerialReceiveOverflow](#)
- [quickSpinIntegerNode ChunkTimestamp](#)
- [quickSpinBooleanNode ChunkModeActive](#)
- [quickSpinIntegerNode ChunkExposureEndLineStatusAll](#)
- [quickSpinEnumerationNode ChunkGainSelector](#)
- [quickSpinEnumerationNode ChunkSelector](#)
- [quickSpinEnumerationNode ChunkBlackLevelSelector](#)
- [quickSpinIntegerNode ChunkWidth](#)
- [quickSpinIntegerNode ChunkImage](#)
- [quickSpinIntegerNode ChunkHeight](#)
- [quickSpinEnumerationNode ChunkPixelFormat](#)
- [quickSpinFloatNode ChunkGain](#)
- [quickSpinIntegerNode ChunkSequencerSetActive](#)
- [quickSpinIntegerNode ChunkCRC](#)
- [quickSpinIntegerNode ChunkOffsetX](#)
- [quickSpinIntegerNode ChunkOffsetY](#)
- [quickSpinBooleanNode ChunkEnable](#)
- [quickSpinIntegerNode ChunkSerialDataLength](#)
- [quickSpinIntegerNode FileAccessOffset](#)
- [quickSpinIntegerNode FileAccessLength](#)
- [quickSpinEnumerationNode FileOperationStatus](#)
- [quickSpinCommandNode FileOperationExecute](#)
- [quickSpinEnumerationNode FileOpenMode](#)
- [quickSpinIntegerNode FileOperationResult](#)
- [quickSpinEnumerationNode FileOperationSelector](#)
- [quickSpinEnumerationNode FileSelector](#)
- [quickSpinIntegerNode FileSize](#)
- [quickSpinEnumerationNode BinningSelector](#)
- [quickSpinIntegerNode PixelDynamicRangeMin](#)
- [quickSpinIntegerNode PixelDynamicRangeMax](#)
- [quickSpinIntegerNode OffsetY](#)
- [quickSpinIntegerNode BinningHorizontal](#)
- [quickSpinIntegerNode Width](#)
- [quickSpinEnumerationNode TestPatternGeneratorSelector](#)
- [quickSpinFloatNode CompressionRatio](#)

- [quickSpinEnumerationNode CompressionSaturationPriority](#)
- [quickSpinBooleanNode ReverseX](#)
- [quickSpinBooleanNode ReverseY](#)
- [quickSpinEnumerationNode TestPattern](#)
- [quickSpinEnumerationNode PixelColorFilter](#)
- [quickSpinIntegerNode WidthMax](#)
- [quickSpinEnumerationNode AdcBitDepth](#)
- [quickSpinIntegerNode BinningVertical](#)
- [quickSpinEnumerationNode DecimationHorizontalMode](#)
- [quickSpinEnumerationNode BinningVerticalMode](#)
- [quickSpinIntegerNode OffsetX](#)
- [quickSpinIntegerNode HeightMax](#)
- [quickSpinIntegerNode DecimationHorizontal](#)
- [quickSpinEnumerationNode PixelSize](#)
- [quickSpinIntegerNode SensorHeight](#)
- [quickSpinEnumerationNode DecimationSelector](#)
- [quickSpinBooleanNode IspEnable](#)
- [quickSpinBooleanNode AdaptiveCompressionEnable](#)
- [quickSpinEnumerationNode ImageCompressionMode](#)
- [quickSpinIntegerNode DecimationVertical](#)
- [quickSpinIntegerNode Height](#)
- [quickSpinEnumerationNode BinningHorizontalMode](#)
- [quickSpinEnumerationNode PixelFormat](#)
- [quickSpinIntegerNode SensorWidth](#)
- [quickSpinEnumerationNode DecimationVerticalMode](#)
- [quickSpinCommandNode TestEventGenerate](#)
- [quickSpinCommandNode TriggerEventTest](#)
- [quickSpinIntegerNode GuiXmlManifestAddress](#)
- [quickSpinIntegerNode Test0001](#)
- [quickSpinBooleanNode V3_3Enable](#)
- [quickSpinEnumerationNode LineMode](#)
- [quickSpinEnumerationNode LineSource](#)
- [quickSpinEnumerationNode LineInputFilterSelector](#)
- [quickSpinBooleanNode UserOutputValue](#)
- [quickSpinIntegerNode UserOutputValueAll](#)
- [quickSpinEnumerationNode UserOutputSelector](#)
- [quickSpinBooleanNode LineStatus](#)
- [quickSpinEnumerationNode LineFormat](#)
- [quickSpinIntegerNode LineStatusAll](#)
- [quickSpinEnumerationNode LineSelector](#)
- [quickSpinEnumerationNode ExposureActiveMode](#)
- [quickSpinBooleanNode LineInverter](#)
- [quickSpinFloatNode LineFilterWidth](#)
- [quickSpinEnumerationNode CounterTriggerActivation](#)
- [quickSpinIntegerNode CounterValue](#)
- [quickSpinEnumerationNode CounterSelector](#)
- [quickSpinIntegerNode CounterValueAtReset](#)
- [quickSpinEnumerationNode CounterStatus](#)
- [quickSpinEnumerationNode CounterTriggerSource](#)
- [quickSpinIntegerNode CounterDelay](#)
- [quickSpinEnumerationNode CounterResetSource](#)
- [quickSpinEnumerationNode CounterEventSource](#)
- [quickSpinEnumerationNode CounterEventActivation](#)
- [quickSpinIntegerNode CounterDuration](#)
- [quickSpinEnumerationNode CounterResetActivation](#)

- [quickSpinEnumerationNode DeviceType](#)
- [quickSpinStringNode DeviceFamilyName](#)
- [quickSpinIntegerNode DeviceSFNCVersionMajor](#)
- [quickSpinIntegerNode DeviceSFNCVersionMinor](#)
- [quickSpinIntegerNode DeviceSFNCVersionSubMinor](#)
- [quickSpinIntegerNode DeviceManifestEntrySelector](#)
- [quickSpinIntegerNode DeviceManifestXMLMajorVersion](#)
- [quickSpinIntegerNode DeviceManifestXMLMinorVersion](#)
- [quickSpinIntegerNode DeviceManifestXMLSubMinorVersion](#)
- [quickSpinIntegerNode DeviceManifestSchemaMajorVersion](#)
- [quickSpinIntegerNode DeviceManifestSchemaMinorVersion](#)
- [quickSpinStringNode DeviceManifestPrimaryURL](#)
- [quickSpinStringNode DeviceManifestSecondaryURL](#)
- [quickSpinIntegerNode DeviceTLVersionSubMinor](#)
- [quickSpinIntegerNode DeviceGenCPVersionMajor](#)
- [quickSpinIntegerNode DeviceGenCPVersionMinor](#)
- [quickSpinIntegerNode DeviceConnectionSelector](#)
- [quickSpinIntegerNode DeviceConnectionSpeed](#)
- [quickSpinEnumerationNode DeviceConnectionStatus](#)
- [quickSpinIntegerNode DeviceLinkSelector](#)
- [quickSpinEnumerationNode DeviceLinkThroughputLimitMode](#)
- [quickSpinIntegerNode DeviceLinkConnectionCount](#)
- [quickSpinEnumerationNode DeviceLinkHeartbeatMode](#)
- [quickSpinFloatNode DeviceLinkHeartbeatTimeout](#)
- [quickSpinFloatNode DeviceLinkCommandTimeout](#)
- [quickSpinIntegerNode DeviceStreamChannelSelector](#)
- [quickSpinEnumerationNode DeviceStreamChannelType](#)
- [quickSpinIntegerNode DeviceStreamChannelLink](#)
- [quickSpinEnumerationNode DeviceStreamChannelEndianness](#)
- [quickSpinIntegerNode DeviceStreamChannelPacketSize](#)
- [quickSpinCommandNode DeviceFeaturePersistenceStart](#)
- [quickSpinCommandNode DeviceFeaturePersistenceEnd](#)
- [quickSpinCommandNode DeviceRegistersStreamingStart](#)
- [quickSpinCommandNode DeviceRegistersStreamingEnd](#)
- [quickSpinCommandNode DeviceRegistersCheck](#)
- [quickSpinBooleanNode DeviceRegistersValid](#)
- [quickSpinEnumerationNode DeviceClockSelector](#)
- [quickSpinFloatNode DeviceClockFrequency](#)
- [quickSpinEnumerationNode DeviceSerialPortSelector](#)
- [quickSpinEnumerationNode DeviceSerialPortBaudRate](#)
- [quickSpinIntegerNode Timestamp](#)
- [quickSpinEnumerationNode SensorTaps](#)
- [quickSpinEnumerationNode SensorDigitizationTaps](#)
- [quickSpinEnumerationNode RegionSelector](#)
- [quickSpinEnumerationNode RegionMode](#)
- [quickSpinEnumerationNode RegionDestination](#)
- [quickSpinEnumerationNode ImageComponentSelector](#)
- [quickSpinBooleanNode ImageComponentEnable](#)
- [quickSpinIntegerNode LinePitch](#)
- [quickSpinEnumerationNode PixelFormatInfoSelector](#)
- [quickSpinIntegerNode PixelFormatInfoID](#)
- [quickSpinEnumerationNode Deinterlacing](#)
- [quickSpinEnumerationNode ImageCompressionRateOption](#)
- [quickSpinIntegerNode ImageCompressionQuality](#)
- [quickSpinFloatNode ImageCompressionBitrate](#)

- quickSpinEnumerationNode ImageCompressionJPEGFormatOption
- quickSpinCommandNode AcquisitionAbort
- quickSpinCommandNode AcquisitionArm
- quickSpinEnumerationNode AcquisitionStatusSelector
- quickSpinBooleanNode AcquisitionStatus
- quickSpinIntegerNode TriggerDivider
- quickSpinIntegerNode TriggerMultiplier
- quickSpinEnumerationNode ExposureTimeMode
- quickSpinEnumerationNode ExposureTimeSelector
- quickSpinEnumerationNode GainAutoBalance
- quickSpinEnumerationNode BlackLevelAuto
- quickSpinEnumerationNode BlackLevelAutoBalance
- quickSpinEnumerationNode WhiteClipSelector
- quickSpinFloatNode WhiteClip
- quickSpinRegisterNode LUTValueAll
- quickSpinIntegerNode UserOutputValueAllMask
- quickSpinCommandNode CounterReset
- quickSpinEnumerationNode TimerSelector
- quickSpinFloatNode TimerDuration
- quickSpinFloatNode TimerDelay
- quickSpinCommandNode TimerReset
- quickSpinFloatNode TimerValue
- quickSpinEnumerationNode TimerStatus
- quickSpinEnumerationNode TimerTriggerSource
- quickSpinEnumerationNode TimerTriggerActivation
- quickSpinEnumerationNode EncoderSelector
- quickSpinEnumerationNode EncoderSourceA
- quickSpinEnumerationNode EncoderSourceB
- quickSpinEnumerationNode EncoderMode
- quickSpinIntegerNode EncoderDivider
- quickSpinEnumerationNode EncoderOutputMode
- quickSpinEnumerationNode EncoderStatus
- quickSpinFloatNode EncoderTimeout
- quickSpinEnumerationNode EncoderResetSource
- quickSpinEnumerationNode EncoderResetActivation
- quickSpinCommandNode EncoderReset
- quickSpinIntegerNode EncoderValue
- quickSpinIntegerNode EncoderValueAtReset
- quickSpinEnumerationNode SoftwareSignalSelector
- quickSpinCommandNode SoftwareSignalPulse
- quickSpinEnumerationNode ActionUnconditionalMode
- quickSpinIntegerNode ActionDeviceKey
- quickSpinIntegerNode ActionQueueSize
- quickSpinIntegerNode ActionSelector
- quickSpinIntegerNode ActionGroupMask
- quickSpinIntegerNode ActionGroupKey
- quickSpinIntegerNode EventAcquisitionTrigger
- quickSpinIntegerNode EventAcquisitionTriggerTimestamp
- quickSpinIntegerNode EventAcquisitionTriggerFrameID
- quickSpinIntegerNode EventAcquisitionStart
- quickSpinIntegerNode EventAcquisitionStartTimestamp
- quickSpinIntegerNode EventAcquisitionStartFrameID
- quickSpinIntegerNode EventAcquisitionEnd
- quickSpinIntegerNode EventAcquisitionEndTimestamp
- quickSpinIntegerNode EventAcquisitionEndFrameID

- [quickSpinIntegerNode EventAcquisitionTransferStart](#)
- [quickSpinIntegerNode EventAcquisitionTransferStartTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionTransferStartFrameID](#)
- [quickSpinIntegerNode EventAcquisitionTransferEnd](#)
- [quickSpinIntegerNode EventAcquisitionTransferEndTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionTransferEndFrameID](#)
- [quickSpinIntegerNode EventAcquisitionError](#)
- [quickSpinIntegerNode EventAcquisitionErrorTimestamp](#)
- [quickSpinIntegerNode EventAcquisitionErrorFrameID](#)
- [quickSpinIntegerNode EventFrameTrigger](#)
- [quickSpinIntegerNode EventFrameTriggerTimestamp](#)
- [quickSpinIntegerNode EventFrameTriggerFrameID](#)
- [quickSpinIntegerNode EventFrameStart](#)
- [quickSpinIntegerNode EventFrameStartTimestamp](#)
- [quickSpinIntegerNode EventFrameStartFrameID](#)
- [quickSpinIntegerNode EventFrameEnd](#)
- [quickSpinIntegerNode EventFrameEndTimestamp](#)
- [quickSpinIntegerNode EventFrameEndFrameID](#)
- [quickSpinIntegerNode EventFrameBurstStart](#)
- [quickSpinIntegerNode EventFrameBurstStartTimestamp](#)
- [quickSpinIntegerNode EventFrameBurstStartFrameID](#)
- [quickSpinIntegerNode EventFrameBurstEnd](#)
- [quickSpinIntegerNode EventFrameBurstEndTimestamp](#)
- [quickSpinIntegerNode EventFrameBurstEndFrameID](#)
- [quickSpinIntegerNode EventFrameTransferStart](#)
- [quickSpinIntegerNode EventFrameTransferStartTimestamp](#)
- [quickSpinIntegerNode EventFrameTransferStartFrameID](#)
- [quickSpinIntegerNode EventFrameTransferEnd](#)
- [quickSpinIntegerNode EventFrameTransferEndTimestamp](#)
- [quickSpinIntegerNode EventFrameTransferEndFrameID](#)
- [quickSpinIntegerNode EventExposureStart](#)
- [quickSpinIntegerNode EventExposureStartTimestamp](#)
- [quickSpinIntegerNode EventExposureStartFrameID](#)
- [quickSpinIntegerNode EventStream0TransferStart](#)
- [quickSpinIntegerNode EventStream0TransferStartTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferStartFrameID](#)
- [quickSpinIntegerNode EventStream0TransferEnd](#)
- [quickSpinIntegerNode EventStream0TransferEndTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferEndFrameID](#)
- [quickSpinIntegerNode EventStream0TransferPause](#)
- [quickSpinIntegerNode EventStream0TransferPauseTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferPauseFrameID](#)
- [quickSpinIntegerNode EventStream0TransferResume](#)
- [quickSpinIntegerNode EventStream0TransferResumeTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferResumeFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBlockStart](#)
- [quickSpinIntegerNode EventStream0TransferBlockStartTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBlockStartFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBlockEnd](#)
- [quickSpinIntegerNode EventStream0TransferBlockEndTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBlockEndFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBlockTrigger](#)
- [quickSpinIntegerNode EventStream0TransferBlockTriggerTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBlockTriggerFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBurstStart](#)

- [quickSpinIntegerNode EventStream0TransferBurstStartTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBurstStartFrameID](#)
- [quickSpinIntegerNode EventStream0TransferBurstEnd](#)
- [quickSpinIntegerNode EventStream0TransferBurstEndTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferBurstEndFrameID](#)
- [quickSpinIntegerNode EventStream0TransferOverflow](#)
- [quickSpinIntegerNode EventStream0TransferOverflowTimestamp](#)
- [quickSpinIntegerNode EventStream0TransferOverflowFrameID](#)
- [quickSpinIntegerNode EventSequencerSetChange](#)
- [quickSpinIntegerNode EventSequencerSetChangeTimestamp](#)
- [quickSpinIntegerNode EventSequencerSetChangeFrameID](#)
- [quickSpinIntegerNode EventCounter0Start](#)
- [quickSpinIntegerNode EventCounter0StartTimestamp](#)
- [quickSpinIntegerNode EventCounter0StartFrameID](#)
- [quickSpinIntegerNode EventCounter1Start](#)
- [quickSpinIntegerNode EventCounter1StartTimestamp](#)
- [quickSpinIntegerNode EventCounter1StartFrameID](#)
- [quickSpinIntegerNode EventCounter0End](#)
- [quickSpinIntegerNode EventCounter0EndTimestamp](#)
- [quickSpinIntegerNode EventCounter0EndFrameID](#)
- [quickSpinIntegerNode EventCounter1End](#)
- [quickSpinIntegerNode EventCounter1EndTimestamp](#)
- [quickSpinIntegerNode EventCounter1EndFrameID](#)
- [quickSpinIntegerNode EventTimer0Start](#)
- [quickSpinIntegerNode EventTimer0StartTimestamp](#)
- [quickSpinIntegerNode EventTimer0StartFrameID](#)
- [quickSpinIntegerNode EventTimer1Start](#)
- [quickSpinIntegerNode EventTimer1StartTimestamp](#)
- [quickSpinIntegerNode EventTimer1StartFrameID](#)
- [quickSpinIntegerNode EventTimer0End](#)
- [quickSpinIntegerNode EventTimer0EndTimestamp](#)
- [quickSpinIntegerNode EventTimer0EndFrameID](#)
- [quickSpinIntegerNode EventTimer1End](#)
- [quickSpinIntegerNode EventTimer1EndTimestamp](#)
- [quickSpinIntegerNode EventTimer1EndFrameID](#)
- [quickSpinIntegerNode EventEncoder0Stopped](#)
- [quickSpinIntegerNode EventEncoder0StoppedTimestamp](#)
- [quickSpinIntegerNode EventEncoder0StoppedFrameID](#)
- [quickSpinIntegerNode EventEncoder1Stopped](#)
- [quickSpinIntegerNode EventEncoder1StoppedTimestamp](#)
- [quickSpinIntegerNode EventEncoder1StoppedFrameID](#)
- [quickSpinIntegerNode EventEncoder0Restarted](#)
- [quickSpinIntegerNode EventEncoder0RestartedTimestamp](#)
- [quickSpinIntegerNode EventEncoder0RestartedFrameID](#)
- [quickSpinIntegerNode EventEncoder1Restarted](#)
- [quickSpinIntegerNode EventEncoder1RestartedTimestamp](#)
- [quickSpinIntegerNode EventEncoder1RestartedFrameID](#)
- [quickSpinIntegerNode EventLine0RisingEdge](#)
- [quickSpinIntegerNode EventLine0RisingEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine0RisingEdgeFrameID](#)
- [quickSpinIntegerNode EventLine1RisingEdge](#)
- [quickSpinIntegerNode EventLine1RisingEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine1RisingEdgeFrameID](#)
- [quickSpinIntegerNode EventLine0FallingEdge](#)
- [quickSpinIntegerNode EventLine0FallingEdgeTimestamp](#)

- [quickSpinIntegerNode EventLine0FallingEdgeFrameID](#)
- [quickSpinIntegerNode EventLine1FallingEdge](#)
- [quickSpinIntegerNode EventLine1FallingEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine1FallingEdgeFrameID](#)
- [quickSpinIntegerNode EventLine0AnyEdge](#)
- [quickSpinIntegerNode EventLine0AnyEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine0AnyEdgeFrameID](#)
- [quickSpinIntegerNode EventLine1AnyEdge](#)
- [quickSpinIntegerNode EventLine1AnyEdgeTimestamp](#)
- [quickSpinIntegerNode EventLine1AnyEdgeFrameID](#)
- [quickSpinIntegerNode EventLinkTrigger0](#)
- [quickSpinIntegerNode EventLinkTrigger0Timestamp](#)
- [quickSpinIntegerNode EventLinkTrigger0FrameID](#)
- [quickSpinIntegerNode EventLinkTrigger1](#)
- [quickSpinIntegerNode EventLinkTrigger1Timestamp](#)
- [quickSpinIntegerNode EventLinkTrigger1FrameID](#)
- [quickSpinIntegerNode EventActionLate](#)
- [quickSpinIntegerNode EventActionLateTimestamp](#)
- [quickSpinIntegerNode EventActionLateFrameID](#)
- [quickSpinIntegerNode EventLinkSpeedChange](#)
- [quickSpinIntegerNode EventLinkSpeedChangeTimestamp](#)
- [quickSpinIntegerNode EventLinkSpeedChangeFrameID](#)
- [quickSpinRegisterNode FileAccessBuffer](#)
- [quickSpinIntegerNode SourceCount](#)
- [quickSpinEnumerationNode SourceSelector](#)
- [quickSpinEnumerationNode TransferSelector](#)
- [quickSpinIntegerNode TransferBurstCount](#)
- [quickSpinCommandNode TransferAbort](#)
- [quickSpinCommandNode TransferPause](#)
- [quickSpinCommandNode TransferResume](#)
- [quickSpinEnumerationNode TransferTriggerSelector](#)
- [quickSpinEnumerationNode TransferTriggerMode](#)
- [quickSpinEnumerationNode TransferTriggerSource](#)
- [quickSpinEnumerationNode TransferTriggerActivation](#)
- [quickSpinEnumerationNode TransferStatusSelector](#)
- [quickSpinBooleanNode TransferStatus](#)
- [quickSpinEnumerationNode TransferComponentSelector](#)
- [quickSpinIntegerNode TransferStreamChannel](#)
- [quickSpinEnumerationNode Scan3dDistanceUnit](#)
- [quickSpinEnumerationNode Scan3dCoordinateSystem](#)
- [quickSpinEnumerationNode Scan3dOutputMode](#)
- [quickSpinEnumerationNode Scan3dCoordinateSystemReference](#)
- [quickSpinEnumerationNode Scan3dCoordinateSelector](#)
- [quickSpinFloatNode Scan3dCoordinateScale](#)
- [quickSpinFloatNode Scan3dCoordinateOffset](#)
- [quickSpinBooleanNode Scan3dInvalidDataFlag](#)
- [quickSpinFloatNode Scan3dInvalidDataValue](#)
- [quickSpinFloatNode Scan3dAxisMin](#)
- [quickSpinFloatNode Scan3dAxisMax](#)
- [quickSpinEnumerationNode Scan3dCoordinateTransformSelector](#)
- [quickSpinFloatNode Scan3dTransformValue](#)
- [quickSpinEnumerationNode Scan3dCoordinateReferenceSelector](#)
- [quickSpinFloatNode Scan3dCoordinateReferenceValue](#)
- [quickSpinIntegerNode ChunkPartSelector](#)
- [quickSpinEnumerationNode ChunkImageComponent](#)

- [quickSpinIntegerNode ChunkPixelDynamicRangeMin](#)
- [quickSpinIntegerNode ChunkPixelDynamicRangeMax](#)
- [quickSpinIntegerNode ChunkTimestampLatchValue](#)
- [quickSpinIntegerNode ChunkLineStatusAll](#)
- [quickSpinEnumerationNode ChunkCounterSelector](#)
- [quickSpinIntegerNode ChunkCounterValue](#)
- [quickSpinEnumerationNode ChunkTimerSelector](#)
- [quickSpinFloatNode ChunkTimerValue](#)
- [quickSpinEnumerationNode ChunkEncoderSelector](#)
- [quickSpinIntegerNode ChunkScanLineSelector](#)
- [quickSpinIntegerNode ChunkEncoderValue](#)
- [quickSpinEnumerationNode ChunkEncoderStatus](#)
- [quickSpinEnumerationNode ChunkExposureTimeSelector](#)
- [quickSpinIntegerNode ChunkLinePitch](#)
- [quickSpinEnumerationNode ChunkSourceID](#)
- [quickSpinEnumerationNode ChunkRegionID](#)
- [quickSpinIntegerNode ChunkTransferBlockID](#)
- [quickSpinEnumerationNode ChunkTransferStreamID](#)
- [quickSpinIntegerNode ChunkTransferQueueCurrentBlockCount](#)
- [quickSpinIntegerNode ChunkStreamChannelID](#)
- [quickSpinEnumerationNode ChunkScan3dDistanceUnit](#)
- [quickSpinEnumerationNode ChunkScan3dOutputMode](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateSystem](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateSystemReference](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateSelector](#)
- [quickSpinFloatNode ChunkScan3dCoordinateScale](#)
- [quickSpinFloatNode ChunkScan3dCoordinateOffset](#)
- [quickSpinBooleanNode ChunkScan3dInvalidDataFlag](#)
- [quickSpinFloatNode ChunkScan3dInvalidDataValue](#)
- [quickSpinFloatNode ChunkScan3dAxisMin](#)
- [quickSpinFloatNode ChunkScan3dAxisMax](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateTransformSelector](#)
- [quickSpinFloatNode ChunkScan3dTransformValue](#)
- [quickSpinEnumerationNode ChunkScan3dCoordinateReferenceSelector](#)
- [quickSpinFloatNode ChunkScan3dCoordinateReferenceValue](#)
- [quickSpinIntegerNode TestPendingAck](#)
- [quickSpinEnumerationNode DeviceTapGeometry](#)
- [quickSpinEnumerationNode GevPhysicalLinkConfiguration](#)
- [quickSpinEnumerationNode GevCurrentPhysicalLinkConfiguration](#)
- [quickSpinIntegerNode GevActiveLinkCount](#)
- [quickSpinBooleanNode GevPAUSEFrameReception](#)
- [quickSpinBooleanNode GevPAUSEFrameTransmission](#)
- [quickSpinEnumerationNode GevIPConfigurationStatus](#)
- [quickSpinIntegerNode GevDiscoveryAckDelay](#)
- [quickSpinEnumerationNode GevGVCPExtendedStatusCodesSelector](#)
- [quickSpinBooleanNode GevGVCPExtendedStatusCodes](#)
- [quickSpinIntegerNode GevPrimaryApplicationSwitchoverKey](#)
- [quickSpinEnumerationNode GevGVSPExtendedIDMode](#)
- [quickSpinIntegerNode GevPrimaryApplicationSocket](#)
- [quickSpinIntegerNode GevPrimaryApplicationIPAddress](#)
- [quickSpinBooleanNode GevSCCFGPacketResendDestination](#)
- [quickSpinBooleanNode GevSCCFGAllInTransmission](#)
- [quickSpinIntegerNode GevSCZoneCount](#)
- [quickSpinIntegerNode GevSCZoneDirectionAll](#)
- [quickSpinBooleanNode GevSCZoneConfigurationLock](#)

- [quickSpinIntegerNode aPAUSEMACCtrlFramesTransmitted](#)
- [quickSpinIntegerNode aPAUSEMACCtrlFramesReceived](#)
- [quickSpinEnumerationNode CIConfiguration](#)
- [quickSpinEnumerationNode CITimeSlotsCount](#)
- [quickSpinEnumerationNode CxpLinkConfigurationStatus](#)
- [quickSpinEnumerationNode CxpLinkConfigurationPreferred](#)
- [quickSpinEnumerationNode CxpLinkConfiguration](#)
- [quickSpinIntegerNode CxpConnectionSelector](#)
- [quickSpinEnumerationNode CxpConnectionTestMode](#)
- [quickSpinIntegerNode CxpConnectionTestErrorCount](#)
- [quickSpinIntegerNode CxpConnectionTestPacketCount](#)
- [quickSpinCommandNode CxpPoCxpAuto](#)
- [quickSpinCommandNode CxpPoCxpTurnOff](#)
- [quickSpinCommandNode CxpPoCxpTripReset](#)
- [quickSpinEnumerationNode CxpPoCxpStatus](#)
- [quickSpinIntegerNode ChunkInferenceFrameId](#)
- [quickSpinIntegerNode ChunkInferenceResult](#)
- [quickSpinFloatNode ChunkInferenceConfidence](#)
- [quickSpinRegisterNode ChunkInferenceBoundingBoxResult](#)

12.2.1 Field Documentation

12.2.1.1 AasRoiEnable

[quickSpinBooleanNode](#) AasRoiEnable

12.2.1.2 AasRoiHeight

[quickSpinIntegerNode](#) AasRoiHeight

12.2.1.3 AasRoiOffsetX

[quickSpinIntegerNode](#) AasRoiOffsetX

12.2.1.4 AasRoiOffsetY

[quickSpinIntegerNode](#) AasRoiOffsetY

12.2.1.5 AasRoiWidth

`quickSpinIntegerNode` AasRoiWidth

12.2.1.6 AcquisitionAbort

`quickSpinCommandNode` AcquisitionAbort

12.2.1.7 AcquisitionArm

`quickSpinCommandNode` AcquisitionArm

12.2.1.8 AcquisitionBurstFrameCount

`quickSpinIntegerNode` AcquisitionBurstFrameCount

12.2.1.9 AcquisitionFrameCount

`quickSpinIntegerNode` AcquisitionFrameCount

12.2.1.10 AcquisitionFrameRate

`quickSpinFloatNode` AcquisitionFrameRate

12.2.1.11 AcquisitionFrameRateEnable

`quickSpinBooleanNode` AcquisitionFrameRateEnable

12.2.1.12 AcquisitionLineRate

`quickSpinFloatNode` AcquisitionLineRate

12.2.1.13 AcquisitionMode

`quickSpinEnumerationNode` AcquisitionMode

12.2.1.14 AcquisitionResultingFrameRate

`quickSpinFloatNode` AcquisitionResultingFrameRate

12.2.1.15 AcquisitionStart

`quickSpinCommandNode` AcquisitionStart

12.2.1.16 AcquisitionStatus

`quickSpinBooleanNode` AcquisitionStatus

12.2.1.17 AcquisitionStatusSelector

`quickSpinEnumerationNode` AcquisitionStatusSelector

12.2.1.18 AcquisitionStop

`quickSpinCommandNode` AcquisitionStop

12.2.1.19 ActionDeviceKey

`quickSpinIntegerNode` ActionDeviceKey

12.2.1.20 ActionGroupKey

`quickSpinIntegerNode` ActionGroupKey

12.2.1.21 ActionGroupMask

`quickSpinIntegerNode` ActionGroupMask

12.2.1.22 ActionQueueSize

`quickSpinIntegerNode` ActionQueueSize

12.2.1.23 ActionSelector

`quickSpinIntegerNode` ActionSelector

12.2.1.24 ActionUnconditionalMode

`quickSpinEnumerationNode` ActionUnconditionalMode

12.2.1.25 AdaptiveCompressionEnable

`quickSpinBooleanNode` AdaptiveCompressionEnable

12.2.1.26 AdcBitDepth

`quickSpinEnumerationNode` AdcBitDepth

12.2.1.27 aPAUSEMACCtrlFramesReceived

`quickSpinIntegerNode` aPAUSEMACCtrlFramesReceived

12.2.1.28 aPAUSEMACCtrlFramesTransmitted

`quickSpinIntegerNode` aPAUSEMACCtrlFramesTransmitted

12.2.1.29 AutoAlgorithmSelector

`quickSpinEnumerationNode` AutoAlgorithmSelector

12.2.1.30 AutoExposureControlLoopDamping

`quickSpinFloatNode` AutoExposureControlLoopDamping

12.2.1.31 AutoExposureControlPriority

`quickSpinEnumerationNode` AutoExposureControlPriority

12.2.1.32 AutoExposureEVCompensation

`quickSpinFloatNode` AutoExposureEVCompensation

12.2.1.33 AutoExposureExposureTimeLowerLimit

`quickSpinFloatNode` AutoExposureExposureTimeLowerLimit

12.2.1.34 AutoExposureExposureTimeUpperLimit

`quickSpinFloatNode` AutoExposureExposureTimeUpperLimit

12.2.1.35 AutoExposureGainLowerLimit

`quickSpinFloatNode` AutoExposureGainLowerLimit

12.2.1.36 AutoExposureGainUpperLimit

`quickSpinFloatNode` AutoExposureGainUpperLimit

12.2.1.37 AutoExposureGreyValueLowerLimit

`quickSpinFloatNode` AutoExposureGreyValueLowerLimit

12.2.1.38 AutoExposureGreyValueUpperLimit

`quickSpinFloatNode` AutoExposureGreyValueUpperLimit

12.2.1.39 AutoExposureLightingMode

`quickSpinEnumerationNode` AutoExposureLightingMode

12.2.1.40 AutoExposureMeteringMode

`quickSpinEnumerationNode` AutoExposureMeteringMode

12.2.1.41 AutoExposureTargetGreyValue

`quickSpinFloatNode` AutoExposureTargetGreyValue

12.2.1.42 AutoExposureTargetGreyValueAuto

`quickSpinEnumerationNode` AutoExposureTargetGreyValueAuto

12.2.1.43 BalanceRatio

`quickSpinFloatNode` BalanceRatio

12.2.1.44 BalanceRatioSelector

`quickSpinEnumerationNode` BalanceRatioSelector

12.2.1.45 BalanceWhiteAuto

`quickSpinEnumerationNode` BalanceWhiteAuto

12.2.1.46 BalanceWhiteAutoDamping

`quickSpinFloatNode` BalanceWhiteAutoDamping

12.2.1.47 BalanceWhiteAutoLowerLimit

`quickSpinFloatNode` BalanceWhiteAutoLowerLimit

12.2.1.48 BalanceWhiteAutoProfile

`quickSpinEnumerationNode` BalanceWhiteAutoProfile

12.2.1.49 BalanceWhiteAutoUpperLimit

`quickSpinFloatNode` BalanceWhiteAutoUpperLimit

12.2.1.50 BinningHorizontal

`quickSpinIntegerNode` BinningHorizontal

12.2.1.51 BinningHorizontalMode

`quickSpinEnumerationNode` BinningHorizontalMode

12.2.1.52 BinningSelector

`quickSpinEnumerationNode` BinningSelector

12.2.1.53 BinningVertical

`quickSpinIntegerNode` BinningVertical

12.2.1.54 BinningVerticalMode

`quickSpinEnumerationNode` BinningVerticalMode

12.2.1.55 BlackLevel

`quickSpinFloatNode` BlackLevel

12.2.1.56 BlackLevelAuto

`quickSpinEnumerationNode` BlackLevelAuto

12.2.1.57 BlackLevelAutoBalance

`quickSpinEnumerationNode` BlackLevelAutoBalance

12.2.1.58 BlackLevelClampingEnable

`quickSpinBooleanNode` BlackLevelClampingEnable

12.2.1.59 BlackLevelRaw

`quickSpinIntegerNode` BlackLevelRaw

12.2.1.60 BlackLevelSelector

`quickSpinEnumerationNode` BlackLevelSelector

12.2.1.61 ChunkBlackLevel

`quickSpinFloatNode` `ChunkBlackLevel`

12.2.1.62 ChunkBlackLevelSelector

`quickSpinEnumerationNode` `ChunkBlackLevelSelector`

12.2.1.63 ChunkCompressionMode

`quickSpinIntegerNode` `ChunkCompressionMode`

12.2.1.64 ChunkCompressionRatio

`quickSpinFloatNode` `ChunkCompressionRatio`

12.2.1.65 ChunkCounterSelector

`quickSpinEnumerationNode` `ChunkCounterSelector`

12.2.1.66 ChunkCounterValue

`quickSpinIntegerNode` `ChunkCounterValue`

12.2.1.67 ChunkCRC

`quickSpinIntegerNode` `ChunkCRC`

12.2.1.68 ChunkEnable

`quickSpinBooleanNode` `ChunkEnable`

12.2.1.69 ChunkEncoderSelector

[quickSpinEnumerationNode](#) ChunkEncoderSelector

12.2.1.70 ChunkEncoderStatus

[quickSpinEnumerationNode](#) ChunkEncoderStatus

12.2.1.71 ChunkEncoderValue

[quickSpinIntegerNode](#) ChunkEncoderValue

12.2.1.72 ChunkExposureEndLineStatusAll

[quickSpinIntegerNode](#) ChunkExposureEndLineStatusAll

12.2.1.73 ChunkExposureTime

[quickSpinFloatNode](#) ChunkExposureTime

12.2.1.74 ChunkExposureTimeSelector

[quickSpinEnumerationNode](#) ChunkExposureTimeSelector

12.2.1.75 ChunkFrameID

[quickSpinIntegerNode](#) ChunkFrameID

12.2.1.76 ChunkGain

[quickSpinFloatNode](#) ChunkGain

12.2.1.77 ChunkGainSelector

[quickSpinEnumerationNode](#) ChunkGainSelector

12.2.1.78 ChunkHeight

[quickSpinIntegerNode](#) ChunkHeight

12.2.1.79 ChunkImage

[quickSpinIntegerNode](#) ChunkImage

12.2.1.80 ChunkImageComponent

[quickSpinEnumerationNode](#) ChunkImageComponent

12.2.1.81 ChunkInferenceBoundingBoxResult

[quickSpinRegisterNode](#) ChunkInferenceBoundingBoxResult

12.2.1.82 ChunkInferenceConfidence

[quickSpinFloatNode](#) ChunkInferenceConfidence

12.2.1.83 ChunkInferenceFrameId

[quickSpinIntegerNode](#) ChunkInferenceFrameId

12.2.1.84 ChunkInferenceResult

[quickSpinIntegerNode](#) ChunkInferenceResult

12.2.1.85 ChunkLinePitch

`quickSpinIntegerNode` ChunkLinePitch

12.2.1.86 ChunkLineStatusAll

`quickSpinIntegerNode` ChunkLineStatusAll

12.2.1.87 ChunkModeActive

`quickSpinBooleanNode` ChunkModeActive

12.2.1.88 ChunkOffsetX

`quickSpinIntegerNode` ChunkOffsetX

12.2.1.89 ChunkOffsetY

`quickSpinIntegerNode` ChunkOffsetY

12.2.1.90 ChunkPartSelector

`quickSpinIntegerNode` ChunkPartSelector

12.2.1.91 ChunkPixelDynamicRangeMax

`quickSpinIntegerNode` ChunkPixelDynamicRangeMax

12.2.1.92 ChunkPixelDynamicRangeMin

`quickSpinIntegerNode` ChunkPixelDynamicRangeMin

12.2.1.93 ChunkPixelFormat

[quickSpinEnumerationNode](#) ChunkPixelFormat

12.2.1.94 ChunkRegionID

[quickSpinEnumerationNode](#) ChunkRegionID

12.2.1.95 ChunkScan3dAxisMax

[quickSpinFloatNode](#) ChunkScan3dAxisMax

12.2.1.96 ChunkScan3dAxisMin

[quickSpinFloatNode](#) ChunkScan3dAxisMin

12.2.1.97 ChunkScan3dCoordinateOffset

[quickSpinFloatNode](#) ChunkScan3dCoordinateOffset

12.2.1.98 ChunkScan3dCoordinateReferenceSelector

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateReferenceSelector

12.2.1.99 ChunkScan3dCoordinateReferenceValue

[quickSpinFloatNode](#) ChunkScan3dCoordinateReferenceValue

12.2.1.100 ChunkScan3dCoordinateScale

[quickSpinFloatNode](#) ChunkScan3dCoordinateScale

12.2.1.101 ChunkScan3dCoordinateSelector

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateSelector

12.2.1.102 ChunkScan3dCoordinateSystem

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateSystem

12.2.1.103 ChunkScan3dCoordinateSystemReference

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateSystemReference

12.2.1.104 ChunkScan3dCoordinateTransformSelector

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateTransformSelector

12.2.1.105 ChunkScan3dDistanceUnit

[quickSpinEnumerationNode](#) ChunkScan3dDistanceUnit

12.2.1.106 ChunkScan3dInvalidDataFlag

[quickSpinBooleanNode](#) ChunkScan3dInvalidDataFlag

12.2.1.107 ChunkScan3dInvalidDataValue

[quickSpinFloatNode](#) ChunkScan3dInvalidDataValue

12.2.1.108 ChunkScan3dOutputMode

[quickSpinEnumerationNode](#) ChunkScan3dOutputMode

12.2.1.109 ChunkScan3dTransformValue

`quickSpinFloatNode` ChunkScan3dTransformValue

12.2.1.110 ChunkScanLineSelector

`quickSpinIntegerNode` ChunkScanLineSelector

12.2.1.111 ChunkSelector

`quickSpinEnumerationNode` ChunkSelector

12.2.1.112 ChunkSequencerSetActive

`quickSpinIntegerNode` ChunkSequencerSetActive

12.2.1.113 ChunkSerialData

`quickSpinStringNode` ChunkSerialData

12.2.1.114 ChunkSerialDataLength

`quickSpinIntegerNode` ChunkSerialDataLength

12.2.1.115 ChunkSerialReceiveOverflow

`quickSpinBooleanNode` ChunkSerialReceiveOverflow

12.2.1.116 ChunkSourceID

`quickSpinEnumerationNode` ChunkSourceID

12.2.1.117 ChunkStreamChannelID

`quickSpinIntegerNode` ChunkStreamChannelID

12.2.1.118 ChunkTimerSelector

`quickSpinEnumerationNode` ChunkTimerSelector

12.2.1.119 ChunkTimerValue

`quickSpinFloatNode` ChunkTimerValue

12.2.1.120 ChunkTimestamp

`quickSpinIntegerNode` ChunkTimestamp

12.2.1.121 ChunkTimestampLatchValue

`quickSpinIntegerNode` ChunkTimestampLatchValue

12.2.1.122 ChunkTransferBlockID

`quickSpinIntegerNode` ChunkTransferBlockID

12.2.1.123 ChunkTransferQueueCurrentBlockCount

`quickSpinIntegerNode` ChunkTransferQueueCurrentBlockCount

12.2.1.124 ChunkTransferStreamID

`quickSpinEnumerationNode` ChunkTransferStreamID

12.2.1.125 ChunkWidth

`quickSpinIntegerNode` ChunkWidth

12.2.1.126 ClConfiguration

`quickSpinEnumerationNode` ClConfiguration

12.2.1.127 ClTimeSlotsCount

`quickSpinEnumerationNode` ClTimeSlotsCount

12.2.1.128 ColorTransformationEnable

`quickSpinBooleanNode` ColorTransformationEnable

12.2.1.129 ColorTransformationSelector

`quickSpinEnumerationNode` ColorTransformationSelector

12.2.1.130 ColorTransformationValue

`quickSpinFloatNode` ColorTransformationValue

12.2.1.131 ColorTransformationValueSelector

`quickSpinEnumerationNode` ColorTransformationValueSelector

12.2.1.132 CompressionRatio

`quickSpinFloatNode` CompressionRatio

12.2.1.133 CompressionSaturationPriority

`quickSpinEnumerationNode` CompressionSaturationPriority

12.2.1.134 CounterDelay

`quickSpinIntegerNode` CounterDelay

12.2.1.135 CounterDuration

`quickSpinIntegerNode` CounterDuration

12.2.1.136 CounterEventActivation

`quickSpinEnumerationNode` CounterEventActivation

12.2.1.137 CounterEventSource

`quickSpinEnumerationNode` CounterEventSource

12.2.1.138 CounterReset

`quickSpinCommandNode` CounterReset

12.2.1.139 CounterResetActivation

`quickSpinEnumerationNode` CounterResetActivation

12.2.1.140 CounterResetSource

`quickSpinEnumerationNode` CounterResetSource

12.2.1.141 CounterSelector

[quickSpinEnumerationNode](#) CounterSelector

12.2.1.142 CounterStatus

[quickSpinEnumerationNode](#) CounterStatus

12.2.1.143 CounterTriggerActivation

[quickSpinEnumerationNode](#) CounterTriggerActivation

12.2.1.144 CounterTriggerSource

[quickSpinEnumerationNode](#) CounterTriggerSource

12.2.1.145 CounterValue

[quickSpinIntegerNode](#) CounterValue

12.2.1.146 CounterValueAtReset

[quickSpinIntegerNode](#) CounterValueAtReset

12.2.1.147 CxpConnectionSelector

[quickSpinIntegerNode](#) CxpConnectionSelector

12.2.1.148 CxpConnectionTestErrorCount

[quickSpinIntegerNode](#) CxpConnectionTestErrorCount

12.2.1.149 CxpConnectionTestMode

[quickSpinEnumerationNode](#) CxpConnectionTestMode

12.2.1.150 CxpConnectionTestPacketCount

[quickSpinIntegerNode](#) CxpConnectionTestPacketCount

12.2.1.151 CxpLinkConfiguration

[quickSpinEnumerationNode](#) CxpLinkConfiguration

12.2.1.152 CxpLinkConfigurationPreferred

[quickSpinEnumerationNode](#) CxpLinkConfigurationPreferred

12.2.1.153 CxpLinkConfigurationStatus

[quickSpinEnumerationNode](#) CxpLinkConfigurationStatus

12.2.1.154 CxpPoCxpAuto

[quickSpinCommandNode](#) CxpPoCxpAuto

12.2.1.155 CxpPoCxpStatus

[quickSpinEnumerationNode](#) CxpPoCxpStatus

12.2.1.156 CxpPoCxpTripReset

[quickSpinCommandNode](#) CxpPoCxpTripReset

12.2.1.157 CxpPoCxpTurnOff

[quickSpinCommandNode](#) CxpPoCxpTurnOff

12.2.1.158 DecimationHorizontal

[quickSpinIntegerNode](#) DecimationHorizontal

12.2.1.159 DecimationHorizontalMode

[quickSpinEnumerationNode](#) DecimationHorizontalMode

12.2.1.160 DecimationSelector

[quickSpinEnumerationNode](#) DecimationSelector

12.2.1.161 DecimationVertical

[quickSpinIntegerNode](#) DecimationVertical

12.2.1.162 DecimationVerticalMode

[quickSpinEnumerationNode](#) DecimationVerticalMode

12.2.1.163 DefectCorrectionMode

[quickSpinEnumerationNode](#) DefectCorrectionMode

12.2.1.164 DefectCorrectStaticEnable

[quickSpinBooleanNode](#) DefectCorrectStaticEnable

12.2.1.165 DefectTableApply

`quickSpinCommandNode` DefectTableApply

12.2.1.166 DefectTableCoordinateX

`quickSpinIntegerNode` DefectTableCoordinateX

12.2.1.167 DefectTableCoordinateY

`quickSpinIntegerNode` DefectTableCoordinateY

12.2.1.168 DefectTableFactoryRestore

`quickSpinCommandNode` DefectTableFactoryRestore

12.2.1.169 DefectTableIndex

`quickSpinIntegerNode` DefectTableIndex

12.2.1.170 DefectTablePixelCount

`quickSpinIntegerNode` DefectTablePixelCount

12.2.1.171 DefectTableSave

`quickSpinCommandNode` DefectTableSave

12.2.1.172 Deinterlacing

`quickSpinEnumerationNode` Deinterlacing

12.2.1.173 DeviceCharacterSet

[quickSpinEnumerationNode](#) DeviceCharacterSet

12.2.1.174 DeviceClockFrequency

[quickSpinFloatNode](#) DeviceClockFrequency

12.2.1.175 DeviceClockSelector

[quickSpinEnumerationNode](#) DeviceClockSelector

12.2.1.176 DeviceConnectionSelector

[quickSpinIntegerNode](#) DeviceConnectionSelector

12.2.1.177 DeviceConnectionSpeed

[quickSpinIntegerNode](#) DeviceConnectionSpeed

12.2.1.178 DeviceConnectionStatus

[quickSpinEnumerationNode](#) DeviceConnectionStatus

12.2.1.179 DeviceEventChannelCount

[quickSpinIntegerNode](#) DeviceEventChannelCount

12.2.1.180 DeviceFamilyName

[quickSpinStringNode](#) DeviceFamilyName

12.2.1.181 DeviceFeaturePersistenceEnd

[quickSpinCommandNode](#) DeviceFeaturePersistenceEnd

12.2.1.182 DeviceFeaturePersistenceStart

[quickSpinCommandNode](#) DeviceFeaturePersistenceStart

12.2.1.183 DeviceFirmwareVersion

[quickSpinStringNode](#) DeviceFirmwareVersion

12.2.1.184 DeviceGenCPVersionMajor

[quickSpinIntegerNode](#) DeviceGenCPVersionMajor

12.2.1.185 DeviceGenCPVersionMinor

[quickSpinIntegerNode](#) DeviceGenCPVersionMinor

12.2.1.186 DeviceID

[quickSpinStringNode](#) DeviceID

12.2.1.187 DeviceIndicatorMode

[quickSpinEnumerationNode](#) DeviceIndicatorMode

12.2.1.188 DeviceLinkBandwidthReserve

[quickSpinFloatNode](#) DeviceLinkBandwidthReserve

12.2.1.189 DeviceLinkCommandTimeout

`quickSpinFloatNode` DeviceLinkCommandTimeout

12.2.1.190 DeviceLinkConnectionCount

`quickSpinIntegerNode` DeviceLinkConnectionCount

12.2.1.191 DeviceLinkCurrentThroughput

`quickSpinIntegerNode` DeviceLinkCurrentThroughput

12.2.1.192 DeviceLinkHeartbeatMode

`quickSpinEnumerationNode` DeviceLinkHeartbeatMode

12.2.1.193 DeviceLinkHeartbeatTimeout

`quickSpinFloatNode` DeviceLinkHeartbeatTimeout

12.2.1.194 DeviceLinkSelector

`quickSpinIntegerNode` DeviceLinkSelector

12.2.1.195 DeviceLinkSpeed

`quickSpinIntegerNode` DeviceLinkSpeed

12.2.1.196 DeviceLinkThroughputLimit

`quickSpinIntegerNode` DeviceLinkThroughputLimit

12.2.1.197 DeviceLinkThroughputLimitMode

[quickSpinEnumerationNode](#) DeviceLinkThroughputLimitMode

12.2.1.198 DeviceManifestEntrySelector

[quickSpinIntegerNode](#) DeviceManifestEntrySelector

12.2.1.199 DeviceManifestPrimaryURL

[quickSpinStringNode](#) DeviceManifestPrimaryURL

12.2.1.200 DeviceManifestSchemaMajorVersion

[quickSpinIntegerNode](#) DeviceManifestSchemaMajorVersion

12.2.1.201 DeviceManifestSchemaMinorVersion

[quickSpinIntegerNode](#) DeviceManifestSchemaMinorVersion

12.2.1.202 DeviceManifestSecondaryURL

[quickSpinStringNode](#) DeviceManifestSecondaryURL

12.2.1.203 DeviceManifestXMLMajorVersion

[quickSpinIntegerNode](#) DeviceManifestXMLMajorVersion

12.2.1.204 DeviceManifestXMLMinorVersion

[quickSpinIntegerNode](#) DeviceManifestXMLMinorVersion

12.2.1.205 DeviceManifestXMLSubMinorVersion

[quickSpinIntegerNode](#) DeviceManifestXMLSubMinorVersion

12.2.1.206 DeviceManufacturerInfo

[quickSpinStringNode](#) DeviceManufacturerInfo

12.2.1.207 DeviceMaxThroughput

[quickSpinIntegerNode](#) DeviceMaxThroughput

12.2.1.208 DeviceModelName

[quickSpinStringNode](#) DeviceModelName

12.2.1.209 DevicePowerSupplySelector

[quickSpinEnumerationNode](#) DevicePowerSupplySelector

12.2.1.210 DeviceRegistersCheck

[quickSpinCommandNode](#) DeviceRegistersCheck

12.2.1.211 DeviceRegistersEndianness

[quickSpinEnumerationNode](#) DeviceRegistersEndianness

12.2.1.212 DeviceRegistersStreamingEnd

[quickSpinCommandNode](#) DeviceRegistersStreamingEnd

12.2.1.213 DeviceRegistersStreamingStart

[quickSpinCommandNode](#) DeviceRegistersStreamingStart

12.2.1.214 DeviceRegistersValid

[quickSpinBooleanNode](#) DeviceRegistersValid

12.2.1.215 DeviceReset

[quickSpinCommandNode](#) DeviceReset

12.2.1.216 DeviceScanType

[quickSpinEnumerationNode](#) DeviceScanType

12.2.1.217 DeviceSerialNumber

[quickSpinStringNode](#) DeviceSerialNumber

12.2.1.218 DeviceSerialPortBaudRate

[quickSpinEnumerationNode](#) DeviceSerialPortBaudRate

12.2.1.219 DeviceSerialPortSelector

[quickSpinEnumerationNode](#) DeviceSerialPortSelector

12.2.1.220 DeviceSFNCVersionMajor

[quickSpinIntegerNode](#) DeviceSFNCVersionMajor

12.2.1.221 DeviceSFNCVersionMinor

[quickSpinIntegerNode](#) DeviceSFNCVersionMinor

12.2.1.222 DeviceSFNCVersionSubMinor

[quickSpinIntegerNode](#) DeviceSFNCVersionSubMinor

12.2.1.223 DeviceStreamChannelCount

[quickSpinIntegerNode](#) DeviceStreamChannelCount

12.2.1.224 DeviceStreamChannelEndianness

[quickSpinEnumerationNode](#) DeviceStreamChannelEndianness

12.2.1.225 DeviceStreamChannelLink

[quickSpinIntegerNode](#) DeviceStreamChannelLink

12.2.1.226 DeviceStreamChannelPacketSize

[quickSpinIntegerNode](#) DeviceStreamChannelPacketSize

12.2.1.227 DeviceStreamChannelSelector

[quickSpinIntegerNode](#) DeviceStreamChannelSelector

12.2.1.228 DeviceStreamChannelType

[quickSpinEnumerationNode](#) DeviceStreamChannelType

12.2.1.229 DeviceTapGeometry

[quickSpinEnumerationNode](#) DeviceTapGeometry

12.2.1.230 DeviceTemperature

[quickSpinFloatNode](#) DeviceTemperature

12.2.1.231 DeviceTemperatureSelector

[quickSpinEnumerationNode](#) DeviceTemperatureSelector

12.2.1.232 DeviceTLType

[quickSpinEnumerationNode](#) DeviceTLType

12.2.1.233 DeviceTLVersionMajor

[quickSpinIntegerNode](#) DeviceTLVersionMajor

12.2.1.234 DeviceTLVersionMinor

[quickSpinIntegerNode](#) DeviceTLVersionMinor

12.2.1.235 DeviceTLVersionSubMinor

[quickSpinIntegerNode](#) DeviceTLVersionSubMinor

12.2.1.236 DeviceType

[quickSpinEnumerationNode](#) DeviceType

12.2.1.237 DeviceUptime

[quickSpinIntegerNode](#) DeviceUptime

12.2.1.238 DeviceUserID

[quickSpinStringNode](#) DeviceUserID

12.2.1.239 DeviceVendorName

[quickSpinStringNode](#) DeviceVendorName

12.2.1.240 DeviceVersion

[quickSpinStringNode](#) DeviceVersion

12.2.1.241 EncoderDivider

[quickSpinIntegerNode](#) EncoderDivider

12.2.1.242 EncoderMode

[quickSpinEnumerationNode](#) EncoderMode

12.2.1.243 EncoderOutputMode

[quickSpinEnumerationNode](#) EncoderOutputMode

12.2.1.244 EncoderReset

[quickSpinCommandNode](#) EncoderReset

12.2.1.245 EncoderResetActivation

[quickSpinEnumerationNode](#) EncoderResetActivation

12.2.1.246 EncoderResetSource

[quickSpinEnumerationNode](#) EncoderResetSource

12.2.1.247 EncoderSelector

[quickSpinEnumerationNode](#) EncoderSelector

12.2.1.248 EncoderSourceA

[quickSpinEnumerationNode](#) EncoderSourceA

12.2.1.249 EncoderSourceB

[quickSpinEnumerationNode](#) EncoderSourceB

12.2.1.250 EncoderStatus

[quickSpinEnumerationNode](#) EncoderStatus

12.2.1.251 EncoderTimeout

[quickSpinFloatNode](#) EncoderTimeout

12.2.1.252 EncoderValue

[quickSpinIntegerNode](#) EncoderValue

12.2.1.253 EncoderValueAtReset

`quickSpinIntegerNode` EncoderValueAtReset

12.2.1.254 EnumerationCount

`quickSpinIntegerNode` EnumerationCount

12.2.1.255 EventAcquisitionEnd

`quickSpinIntegerNode` EventAcquisitionEnd

12.2.1.256 EventAcquisitionEndFrameID

`quickSpinIntegerNode` EventAcquisitionEndFrameID

12.2.1.257 EventAcquisitionEndTimestamp

`quickSpinIntegerNode` EventAcquisitionEndTimestamp

12.2.1.258 EventAcquisitionError

`quickSpinIntegerNode` EventAcquisitionError

12.2.1.259 EventAcquisitionErrorFrameID

`quickSpinIntegerNode` EventAcquisitionErrorFrameID

12.2.1.260 EventAcquisitionErrorTimestamp

`quickSpinIntegerNode` EventAcquisitionErrorTimestamp

12.2.1.261 EventAcquisitionStart

`quickSpinIntegerNode` EventAcquisitionStart

12.2.1.262 EventAcquisitionStartFrameID

`quickSpinIntegerNode` EventAcquisitionStartFrameID

12.2.1.263 EventAcquisitionStartTimestamp

`quickSpinIntegerNode` EventAcquisitionStartTimestamp

12.2.1.264 EventAcquisitionTransferEnd

`quickSpinIntegerNode` EventAcquisitionTransferEnd

12.2.1.265 EventAcquisitionTransferEndFrameID

`quickSpinIntegerNode` EventAcquisitionTransferEndFrameID

12.2.1.266 EventAcquisitionTransferEndTimestamp

`quickSpinIntegerNode` EventAcquisitionTransferEndTimestamp

12.2.1.267 EventAcquisitionTransferStart

`quickSpinIntegerNode` EventAcquisitionTransferStart

12.2.1.268 EventAcquisitionTransferStartFrameID

`quickSpinIntegerNode` EventAcquisitionTransferStartFrameID

12.2.1.269 EventAcquisitionTransferStartTimestamp

`quickSpinIntegerNode` EventAcquisitionTransferStartTimestamp

12.2.1.270 EventAcquisitionTrigger

`quickSpinIntegerNode` EventAcquisitionTrigger

12.2.1.271 EventAcquisitionTriggerFrameID

`quickSpinIntegerNode` EventAcquisitionTriggerFrameID

12.2.1.272 EventAcquisitionTriggerTimestamp

`quickSpinIntegerNode` EventAcquisitionTriggerTimestamp

12.2.1.273 EventActionLate

`quickSpinIntegerNode` EventActionLate

12.2.1.274 EventActionLateFrameID

`quickSpinIntegerNode` EventActionLateFrameID

12.2.1.275 EventActionLateTimestamp

`quickSpinIntegerNode` EventActionLateTimestamp

12.2.1.276 EventCounter0End

`quickSpinIntegerNode` EventCounter0End

12.2.1.277 EventCounter0EndFrameID

`quickSpinIntegerNode` EventCounter0EndFrameID

12.2.1.278 EventCounter0EndTimestamp

`quickSpinIntegerNode` EventCounter0EndTimestamp

12.2.1.279 EventCounter0Start

`quickSpinIntegerNode` EventCounter0Start

12.2.1.280 EventCounter0StartFrameID

`quickSpinIntegerNode` EventCounter0StartFrameID

12.2.1.281 EventCounter0StartTimestamp

`quickSpinIntegerNode` EventCounter0StartTimestamp

12.2.1.282 EventCounter1End

`quickSpinIntegerNode` EventCounter1End

12.2.1.283 EventCounter1EndFrameID

`quickSpinIntegerNode` EventCounter1EndFrameID

12.2.1.284 EventCounter1EndTimestamp

`quickSpinIntegerNode` EventCounter1EndTimestamp

12.2.1.285 EventCounter1Start

[quickSpinIntegerNode](#) EventCounter1Start

12.2.1.286 EventCounter1StartFrameID

[quickSpinIntegerNode](#) EventCounter1StartFrameID

12.2.1.287 EventCounter1StartTimestamp

[quickSpinIntegerNode](#) EventCounter1StartTimestamp

12.2.1.288 EventEncoder0Restarted

[quickSpinIntegerNode](#) EventEncoder0Restarted

12.2.1.289 EventEncoder0RestartedFrameID

[quickSpinIntegerNode](#) EventEncoder0RestartedFrameID

12.2.1.290 EventEncoder0RestartedTimestamp

[quickSpinIntegerNode](#) EventEncoder0RestartedTimestamp

12.2.1.291 EventEncoder0Stopped

[quickSpinIntegerNode](#) EventEncoder0Stopped

12.2.1.292 EventEncoder0StoppedFrameID

[quickSpinIntegerNode](#) EventEncoder0StoppedFrameID

12.2.1.293 EventEncoder0StoppedTimestamp

`quickSpinIntegerNode` EventEncoder0StoppedTimestamp

12.2.1.294 EventEncoder1Restarted

`quickSpinIntegerNode` EventEncoder1Restarted

12.2.1.295 EventEncoder1RestartedFrameID

`quickSpinIntegerNode` EventEncoder1RestartedFrameID

12.2.1.296 EventEncoder1RestartedTimestamp

`quickSpinIntegerNode` EventEncoder1RestartedTimestamp

12.2.1.297 EventEncoder1Stopped

`quickSpinIntegerNode` EventEncoder1Stopped

12.2.1.298 EventEncoder1StoppedFrameID

`quickSpinIntegerNode` EventEncoder1StoppedFrameID

12.2.1.299 EventEncoder1StoppedTimestamp

`quickSpinIntegerNode` EventEncoder1StoppedTimestamp

12.2.1.300 EventError

`quickSpinIntegerNode` EventError

12.2.1.301 EventErrorCode

`quickSpinIntegerNode` EventErrorCode

12.2.1.302 EventErrorFrameID

`quickSpinIntegerNode` EventErrorFrameID

12.2.1.303 EventErrorTimestamp

`quickSpinIntegerNode` EventErrorTimestamp

12.2.1.304 EventExposureEnd

`quickSpinIntegerNode` EventExposureEnd

12.2.1.305 EventExposureEndFrameID

`quickSpinIntegerNode` EventExposureEndFrameID

12.2.1.306 EventExposureEndTimestamp

`quickSpinIntegerNode` EventExposureEndTimestamp

12.2.1.307 EventExposureStart

`quickSpinIntegerNode` EventExposureStart

12.2.1.308 EventExposureStartFrameID

`quickSpinIntegerNode` EventExposureStartFrameID

12.2.1.309 EventExposureStartTimestamp

[quickSpinIntegerNode](#) EventExposureStartTimestamp

12.2.1.310 EventFrameBurstEnd

[quickSpinIntegerNode](#) EventFrameBurstEnd

12.2.1.311 EventFrameBurstEndFrameID

[quickSpinIntegerNode](#) EventFrameBurstEndFrameID

12.2.1.312 EventFrameBurstEndTimestamp

[quickSpinIntegerNode](#) EventFrameBurstEndTimestamp

12.2.1.313 EventFrameBurstStart

[quickSpinIntegerNode](#) EventFrameBurstStart

12.2.1.314 EventFrameBurstStartFrameID

[quickSpinIntegerNode](#) EventFrameBurstStartFrameID

12.2.1.315 EventFrameBurstStartTimestamp

[quickSpinIntegerNode](#) EventFrameBurstStartTimestamp

12.2.1.316 EventFrameEnd

[quickSpinIntegerNode](#) EventFrameEnd

12.2.1.317 EventFrameEndFrameID

`quickSpinIntegerNode` EventFrameEndFrameID

12.2.1.318 EventFrameEndTimestamp

`quickSpinIntegerNode` EventFrameEndTimestamp

12.2.1.319 EventFrameStart

`quickSpinIntegerNode` EventFrameStart

12.2.1.320 EventFrameStartFrameID

`quickSpinIntegerNode` EventFrameStartFrameID

12.2.1.321 EventFrameStartTimestamp

`quickSpinIntegerNode` EventFrameStartTimestamp

12.2.1.322 EventFrameTransferEnd

`quickSpinIntegerNode` EventFrameTransferEnd

12.2.1.323 EventFrameTransferEndFrameID

`quickSpinIntegerNode` EventFrameTransferEndFrameID

12.2.1.324 EventFrameTransferEndTimestamp

`quickSpinIntegerNode` EventFrameTransferEndTimestamp

12.2.1.325 EventFrameTransferStart

`quickSpinIntegerNode` EventFrameTransferStart

12.2.1.326 EventFrameTransferStartFrameID

`quickSpinIntegerNode` EventFrameTransferStartFrameID

12.2.1.327 EventFrameTransferStartTimestamp

`quickSpinIntegerNode` EventFrameTransferStartTimestamp

12.2.1.328 EventFrameTrigger

`quickSpinIntegerNode` EventFrameTrigger

12.2.1.329 EventFrameTriggerFrameID

`quickSpinIntegerNode` EventFrameTriggerFrameID

12.2.1.330 EventFrameTriggerTimestamp

`quickSpinIntegerNode` EventFrameTriggerTimestamp

12.2.1.331 EventLine0AnyEdge

`quickSpinIntegerNode` EventLine0AnyEdge

12.2.1.332 EventLine0AnyEdgeFrameID

`quickSpinIntegerNode` EventLine0AnyEdgeFrameID

12.2.1.333 EventLine0AnyEdgeTimestamp

`quickSpinIntegerNode` EventLine0AnyEdgeTimestamp

12.2.1.334 EventLine0FallingEdge

`quickSpinIntegerNode` EventLine0FallingEdge

12.2.1.335 EventLine0FallingEdgeFrameID

`quickSpinIntegerNode` EventLine0FallingEdgeFrameID

12.2.1.336 EventLine0FallingEdgeTimestamp

`quickSpinIntegerNode` EventLine0FallingEdgeTimestamp

12.2.1.337 EventLine0RisingEdge

`quickSpinIntegerNode` EventLine0RisingEdge

12.2.1.338 EventLine0RisingEdgeFrameID

`quickSpinIntegerNode` EventLine0RisingEdgeFrameID

12.2.1.339 EventLine0RisingEdgeTimestamp

`quickSpinIntegerNode` EventLine0RisingEdgeTimestamp

12.2.1.340 EventLine1AnyEdge

`quickSpinIntegerNode` EventLine1AnyEdge

12.2.1.341 EventLine1AnyEdgeFrameID

`quickSpinIntegerNode` EventLine1AnyEdgeFrameID

12.2.1.342 EventLine1AnyEdgeTimestamp

`quickSpinIntegerNode` EventLine1AnyEdgeTimestamp

12.2.1.343 EventLine1FallingEdge

`quickSpinIntegerNode` EventLine1FallingEdge

12.2.1.344 EventLine1FallingEdgeFrameID

`quickSpinIntegerNode` EventLine1FallingEdgeFrameID

12.2.1.345 EventLine1FallingEdgeTimestamp

`quickSpinIntegerNode` EventLine1FallingEdgeTimestamp

12.2.1.346 EventLine1RisingEdge

`quickSpinIntegerNode` EventLine1RisingEdge

12.2.1.347 EventLine1RisingEdgeFrameID

`quickSpinIntegerNode` EventLine1RisingEdgeFrameID

12.2.1.348 EventLine1RisingEdgeTimestamp

`quickSpinIntegerNode` EventLine1RisingEdgeTimestamp

12.2.1.349 EventLinkSpeedChange

[quickSpinIntegerNode](#) EventLinkSpeedChange

12.2.1.350 EventLinkSpeedChangeFrameID

[quickSpinIntegerNode](#) EventLinkSpeedChangeFrameID

12.2.1.351 EventLinkSpeedChangeTimestamp

[quickSpinIntegerNode](#) EventLinkSpeedChangeTimestamp

12.2.1.352 EventLinkTrigger0

[quickSpinIntegerNode](#) EventLinkTrigger0

12.2.1.353 EventLinkTrigger0FrameID

[quickSpinIntegerNode](#) EventLinkTrigger0FrameID

12.2.1.354 EventLinkTrigger0Timestamp

[quickSpinIntegerNode](#) EventLinkTrigger0Timestamp

12.2.1.355 EventLinkTrigger1

[quickSpinIntegerNode](#) EventLinkTrigger1

12.2.1.356 EventLinkTrigger1FrameID

[quickSpinIntegerNode](#) EventLinkTrigger1FrameID

12.2.1.357 EventLinkTrigger1Timestamp

[quickSpinIntegerNode](#) EventLinkTrigger1Timestamp

12.2.1.358 EventNotification

[quickSpinEnumerationNode](#) EventNotification

12.2.1.359 EventSelector

[quickSpinEnumerationNode](#) EventSelector

12.2.1.360 EventSequencerSetChange

[quickSpinIntegerNode](#) EventSequencerSetChange

12.2.1.361 EventSequencerSetChangeFrameID

[quickSpinIntegerNode](#) EventSequencerSetChangeFrameID

12.2.1.362 EventSequencerSetChangeTimestamp

[quickSpinIntegerNode](#) EventSequencerSetChangeTimestamp

12.2.1.363 EventSerialData

[quickSpinStringNode](#) EventSerialData

12.2.1.364 EventSerialDataLength

[quickSpinIntegerNode](#) EventSerialDataLength

12.2.1.365 EventSerialPortReceive

`quickSpinIntegerNode` EventSerialPortReceive

12.2.1.366 EventSerialPortReceiveTimestamp

`quickSpinIntegerNode` EventSerialPortReceiveTimestamp

12.2.1.367 EventSerialReceiveOverflow

`quickSpinBooleanNode` EventSerialReceiveOverflow

12.2.1.368 EventStream0TransferBlockEnd

`quickSpinIntegerNode` EventStream0TransferBlockEnd

12.2.1.369 EventStream0TransferBlockEndFrameID

`quickSpinIntegerNode` EventStream0TransferBlockEndFrameID

12.2.1.370 EventStream0TransferBlockEndTimestamp

`quickSpinIntegerNode` EventStream0TransferBlockEndTimestamp

12.2.1.371 EventStream0TransferBlockStart

`quickSpinIntegerNode` EventStream0TransferBlockStart

12.2.1.372 EventStream0TransferBlockStartFrameID

`quickSpinIntegerNode` EventStream0TransferBlockStartFrameID

12.2.1.373 EventStream0TransferBlockStartTimestamp

`quickSpinIntegerNode` EventStream0TransferBlockStartTimestamp

12.2.1.374 EventStream0TransferBlockTrigger

`quickSpinIntegerNode` EventStream0TransferBlockTrigger

12.2.1.375 EventStream0TransferBlockTriggerFrameID

`quickSpinIntegerNode` EventStream0TransferBlockTriggerFrameID

12.2.1.376 EventStream0TransferBlockTriggerTimestamp

`quickSpinIntegerNode` EventStream0TransferBlockTriggerTimestamp

12.2.1.377 EventStream0TransferBurstEnd

`quickSpinIntegerNode` EventStream0TransferBurstEnd

12.2.1.378 EventStream0TransferBurstEndFrameID

`quickSpinIntegerNode` EventStream0TransferBurstEndFrameID

12.2.1.379 EventStream0TransferBurstEndTimestamp

`quickSpinIntegerNode` EventStream0TransferBurstEndTimestamp

12.2.1.380 EventStream0TransferBurstStart

`quickSpinIntegerNode` EventStream0TransferBurstStart

12.2.1.381 EventStream0TransferBurstStartFrameID

`quickSpinIntegerNode` EventStream0TransferBurstStartFrameID

12.2.1.382 EventStream0TransferBurstStartTimestamp

`quickSpinIntegerNode` EventStream0TransferBurstStartTimestamp

12.2.1.383 EventStream0TransferEnd

`quickSpinIntegerNode` EventStream0TransferEnd

12.2.1.384 EventStream0TransferEndFrameID

`quickSpinIntegerNode` EventStream0TransferEndFrameID

12.2.1.385 EventStream0TransferEndTimestamp

`quickSpinIntegerNode` EventStream0TransferEndTimestamp

12.2.1.386 EventStream0TransferOverflow

`quickSpinIntegerNode` EventStream0TransferOverflow

12.2.1.387 EventStream0TransferOverflowFrameID

`quickSpinIntegerNode` EventStream0TransferOverflowFrameID

12.2.1.388 EventStream0TransferOverflowTimestamp

`quickSpinIntegerNode` EventStream0TransferOverflowTimestamp

12.2.1.389 EventStream0TransferPause

[quickSpinIntegerNode](#) EventStream0TransferPause

12.2.1.390 EventStream0TransferPauseFrameID

[quickSpinIntegerNode](#) EventStream0TransferPauseFrameID

12.2.1.391 EventStream0TransferPauseTimestamp

[quickSpinIntegerNode](#) EventStream0TransferPauseTimestamp

12.2.1.392 EventStream0TransferResume

[quickSpinIntegerNode](#) EventStream0TransferResume

12.2.1.393 EventStream0TransferResumeFrameID

[quickSpinIntegerNode](#) EventStream0TransferResumeFrameID

12.2.1.394 EventStream0TransferResumeTimestamp

[quickSpinIntegerNode](#) EventStream0TransferResumeTimestamp

12.2.1.395 EventStream0TransferStart

[quickSpinIntegerNode](#) EventStream0TransferStart

12.2.1.396 EventStream0TransferStartFrameID

[quickSpinIntegerNode](#) EventStream0TransferStartFrameID

12.2.1.397 EventStream0TransferStartTimestamp

[quickSpinIntegerNode](#) EventStream0TransferStartTimestamp

12.2.1.398 EventTest

[quickSpinIntegerNode](#) EventTest

12.2.1.399 EventTestTimestamp

[quickSpinIntegerNode](#) EventTestTimestamp

12.2.1.400 EventTimer0End

[quickSpinIntegerNode](#) EventTimer0End

12.2.1.401 EventTimer0EndFrameID

[quickSpinIntegerNode](#) EventTimer0EndFrameID

12.2.1.402 EventTimer0EndTimestamp

[quickSpinIntegerNode](#) EventTimer0EndTimestamp

12.2.1.403 EventTimer0Start

[quickSpinIntegerNode](#) EventTimer0Start

12.2.1.404 EventTimer0StartFrameID

[quickSpinIntegerNode](#) EventTimer0StartFrameID

12.2.1.405 EventTimer0StartTimestamp

[quickSpinIntegerNode](#) EventTimer0StartTimestamp

12.2.1.406 EventTimer1End

[quickSpinIntegerNode](#) EventTimer1End

12.2.1.407 EventTimer1EndFrameID

[quickSpinIntegerNode](#) EventTimer1EndFrameID

12.2.1.408 EventTimer1EndTimestamp

[quickSpinIntegerNode](#) EventTimer1EndTimestamp

12.2.1.409 EventTimer1Start

[quickSpinIntegerNode](#) EventTimer1Start

12.2.1.410 EventTimer1StartFrameID

[quickSpinIntegerNode](#) EventTimer1StartFrameID

12.2.1.411 EventTimer1StartTimestamp

[quickSpinIntegerNode](#) EventTimer1StartTimestamp

12.2.1.412 ExposureActiveMode

[quickSpinEnumerationNode](#) ExposureActiveMode

12.2.1.413 ExposureAuto

[quickSpinEnumerationNode](#) ExposureAuto

12.2.1.414 ExposureMode

[quickSpinEnumerationNode](#) ExposureMode

12.2.1.415 ExposureTime

[quickSpinFloatNode](#) ExposureTime

12.2.1.416 ExposureTimeMode

[quickSpinEnumerationNode](#) ExposureTimeMode

12.2.1.417 ExposureTimeSelector

[quickSpinEnumerationNode](#) ExposureTimeSelector

12.2.1.418 FactoryReset

[quickSpinCommandNode](#) FactoryReset

12.2.1.419 FileAccessBuffer

[quickSpinRegisterNode](#) FileAccessBuffer

12.2.1.420 FileAccessLength

[quickSpinIntegerNode](#) FileAccessLength

12.2.1.421 FileAccessOffset

`quickSpinIntegerNode` FileAccessOffset

12.2.1.422 FileOpenMode

`quickSpinEnumerationNode` FileOpenMode

12.2.1.423 FileOperationExecute

`quickSpinCommandNode` FileOperationExecute

12.2.1.424 FileOperationResult

`quickSpinIntegerNode` FileOperationResult

12.2.1.425 FileOperationSelector

`quickSpinEnumerationNode` FileOperationSelector

12.2.1.426 FileOperationStatus

`quickSpinEnumerationNode` FileOperationStatus

12.2.1.427 FileSelector

`quickSpinEnumerationNode` FileSelector

12.2.1.428 FileSize

`quickSpinIntegerNode` FileSize

12.2.1.429 Gain

[quickSpinFloatNode](#) Gain

12.2.1.430 GainAuto

[quickSpinEnumerationNode](#) GainAuto

12.2.1.431 GainAutoBalance

[quickSpinEnumerationNode](#) GainAutoBalance

12.2.1.432 GainSelector

[quickSpinEnumerationNode](#) GainSelector

12.2.1.433 Gamma

[quickSpinFloatNode](#) Gamma

12.2.1.434 GammaEnable

[quickSpinBooleanNode](#) GammaEnable

12.2.1.435 GevActiveLinkCount

[quickSpinIntegerNode](#) GevActiveLinkCount

12.2.1.436 GevCCP

[quickSpinEnumerationNode](#) GevCCP

12.2.1.437 GevCurrentDefaultGateway

`quickSpinIntegerNode` `GevCurrentDefaultGateway`

12.2.1.438 GevCurrentIPAddress

`quickSpinIntegerNode` `GevCurrentIPAddress`

12.2.1.439 GevCurrentIPConfigurationDHCP

`quickSpinBooleanNode` `GevCurrentIPConfigurationDHCP`

12.2.1.440 GevCurrentIPConfigurationLLA

`quickSpinBooleanNode` `GevCurrentIPConfigurationLLA`

12.2.1.441 GevCurrentIPConfigurationPersistentIP

`quickSpinBooleanNode` `GevCurrentIPConfigurationPersistentIP`

12.2.1.442 GevCurrentPhysicalLinkConfiguration

`quickSpinEnumerationNode` `GevCurrentPhysicalLinkConfiguration`

12.2.1.443 GevCurrentSubnetMask

`quickSpinIntegerNode` `GevCurrentSubnetMask`

12.2.1.444 GevDiscoveryAckDelay

`quickSpinIntegerNode` `GevDiscoveryAckDelay`

12.2.1.445 GevFirstURL

`quickSpinStringNode` `GevFirstURL`

12.2.1.446 GevGVCPExtendedStatusCodes

`quickSpinBooleanNode` `GevGVCPExtendedStatusCodes`

12.2.1.447 GevGVCPExtendedStatusCodesSelector

`quickSpinEnumerationNode` `GevGVCPExtendedStatusCodesSelector`

12.2.1.448 GevGVCPHeartbeatDisable

`quickSpinBooleanNode` `GevGVCPHeartbeatDisable`

12.2.1.449 GevGVCPPendingAck

`quickSpinBooleanNode` `GevGVCPPendingAck`

12.2.1.450 GevGVCPPendingTimeout

`quickSpinIntegerNode` `GevGVCPPendingTimeout`

12.2.1.451 GevGVSPExtendedIDMode

`quickSpinEnumerationNode` `GevGVSPExtendedIDMode`

12.2.1.452 GevHeartbeatTimeout

`quickSpinIntegerNode` `GevHeartbeatTimeout`

12.2.1.453 GevIEEE1588

`quickSpinBooleanNode` `GevIEEE1588`

12.2.1.454 GevIEEE1588ClockAccuracy

`quickSpinEnumerationNode` `GevIEEE1588ClockAccuracy`

12.2.1.455 GevIEEE1588Mode

`quickSpinEnumerationNode` `GevIEEE1588Mode`

12.2.1.456 GevIEEE1588Status

`quickSpinEnumerationNode` `GevIEEE1588Status`

12.2.1.457 GevInterfaceSelector

`quickSpinIntegerNode` `GevInterfaceSelector`

12.2.1.458 GevIPConfigurationStatus

`quickSpinEnumerationNode` `GevIPConfigurationStatus`

12.2.1.459 GevMACAddress

`quickSpinIntegerNode` `GevMACAddress`

12.2.1.460 GevMCDA

`quickSpinIntegerNode` `GevMCDA`

12.2.1.461 GevMCPHostPort

`quickSpinIntegerNode` GevMCPHostPort

12.2.1.462 GevMCRC

`quickSpinIntegerNode` GevMCRC

12.2.1.463 GevMCSP

`quickSpinIntegerNode` GevMCSP

12.2.1.464 GevMCTT

`quickSpinIntegerNode` GevMCTT

12.2.1.465 GevNumberOfInterfaces

`quickSpinIntegerNode` GevNumberOfInterfaces

12.2.1.466 GevPAUSEFrameReception

`quickSpinBooleanNode` GevPAUSEFrameReception

12.2.1.467 GevPAUSEFrameTransmission

`quickSpinBooleanNode` GevPAUSEFrameTransmission

12.2.1.468 GevPersistentDefaultGateway

`quickSpinIntegerNode` GevPersistentDefaultGateway

12.2.1.469 GevPersistentIPAddress

`quickSpinIntegerNode` GevPersistentIPAddress

12.2.1.470 GevPersistentSubnetMask

`quickSpinIntegerNode` GevPersistentSubnetMask

12.2.1.471 GevPhysicalLinkConfiguration

`quickSpinEnumerationNode` GevPhysicalLinkConfiguration

12.2.1.472 GevPrimaryApplicationIPAddress

`quickSpinIntegerNode` GevPrimaryApplicationIPAddress

12.2.1.473 GevPrimaryApplicationSocket

`quickSpinIntegerNode` GevPrimaryApplicationSocket

12.2.1.474 GevPrimaryApplicationSwitchoverKey

`quickSpinIntegerNode` GevPrimaryApplicationSwitchoverKey

12.2.1.475 GevSCCFGAllInTransmission

`quickSpinBooleanNode` GevSCCFGAllInTransmission

12.2.1.476 GevSCCFGExtendedChunkData

`quickSpinBooleanNode` GevSCCFGExtendedChunkData

12.2.1.477 GevSCCFGPacketResendDestination

[quickSpinBooleanNode](#) GevSCCFGPacketResendDestination

12.2.1.478 GevSCCFGUnconditionalStreaming

[quickSpinBooleanNode](#) GevSCCFGUnconditionalStreaming

12.2.1.479 GevSCDA

[quickSpinIntegerNode](#) GevSCDA

12.2.1.480 GevSCPD

[quickSpinIntegerNode](#) GevSCPD

12.2.1.481 GevSCPDirection

[quickSpinIntegerNode](#) GevSCPDirection

12.2.1.482 GevSCPHostPort

[quickSpinIntegerNode](#) GevSCPHostPort

12.2.1.483 GevSCPInterfaceIndex

[quickSpinIntegerNode](#) GevSCPInterfaceIndex

12.2.1.484 GevSCPSBigEndian

[quickSpinBooleanNode](#) GevSCPSBigEndian

12.2.1.485 GevSCPSDoNotFragment

[quickSpinBooleanNode](#) GevSCPSDoNotFragment

12.2.1.486 GevSCPSFireTestPacket

[quickSpinBooleanNode](#) GevSCPSFireTestPacket

12.2.1.487 GevSCSPPacketSize

[quickSpinIntegerNode](#) GevSCSPPacketSize

12.2.1.488 GevSCSP

[quickSpinIntegerNode](#) GevSCSP

12.2.1.489 GevSCZoneConfigurationLock

[quickSpinBooleanNode](#) GevSCZoneConfigurationLock

12.2.1.490 GevSCZoneCount

[quickSpinIntegerNode](#) GevSCZoneCount

12.2.1.491 GevSCZoneDirectionAll

[quickSpinIntegerNode](#) GevSCZoneDirectionAll

12.2.1.492 GevSecondURL

[quickSpinStringNode](#) GevSecondURL

12.2.1.493 GevStreamChannelSelector

`quickSpinIntegerNode` `GevStreamChannelSelector`

12.2.1.494 GevSupportedOption

`quickSpinBooleanNode` `GevSupportedOption`

12.2.1.495 GevSupportedOptionSelector

`quickSpinEnumerationNode` `GevSupportedOptionSelector`

12.2.1.496 GevTimestampTickFrequency

`quickSpinIntegerNode` `GevTimestampTickFrequency`

12.2.1.497 GuiXmlManifestAddress

`quickSpinIntegerNode` `GuiXmlManifestAddress`

12.2.1.498 Height

`quickSpinIntegerNode` `Height`

12.2.1.499 HeightMax

`quickSpinIntegerNode` `HeightMax`

12.2.1.500 ImageComponentEnable

`quickSpinBooleanNode` `ImageComponentEnable`

12.2.1.501 ImageComponentSelector

`quickSpinEnumerationNode` ImageComponentSelector

12.2.1.502 ImageCompressionBitrate

`quickSpinFloatNode` ImageCompressionBitrate

12.2.1.503 ImageCompressionJPEGFormatOption

`quickSpinEnumerationNode` ImageCompressionJPEGFormatOption

12.2.1.504 ImageCompressionMode

`quickSpinEnumerationNode` ImageCompressionMode

12.2.1.505 ImageCompressionQuality

`quickSpinIntegerNode` ImageCompressionQuality

12.2.1.506 ImageCompressionRateOption

`quickSpinEnumerationNode` ImageCompressionRateOption

12.2.1.507 IspEnable

`quickSpinBooleanNode` IspEnable

12.2.1.508 LineFilterWidth

`quickSpinFloatNode` LineFilterWidth

12.2.1.509 LineFormat

[quickSpinEnumerationNode](#) LineFormat

12.2.1.510 LineInputFilterSelector

[quickSpinEnumerationNode](#) LineInputFilterSelector

12.2.1.511 LineInverter

[quickSpinBooleanNode](#) LineInverter

12.2.1.512 LineMode

[quickSpinEnumerationNode](#) LineMode

12.2.1.513 LinePitch

[quickSpinIntegerNode](#) LinePitch

12.2.1.514 LineSelector

[quickSpinEnumerationNode](#) LineSelector

12.2.1.515 LineSource

[quickSpinEnumerationNode](#) LineSource

12.2.1.516 LineStatus

[quickSpinBooleanNode](#) LineStatus

12.2.1.517 LineStatusAll

`quickSpinIntegerNode` LineStatusAll

12.2.1.518 LinkErrorCount

`quickSpinIntegerNode` LinkErrorCount

12.2.1.519 LinkUptime

`quickSpinIntegerNode` LinkUptime

12.2.1.520 LogicBlockLUTInputActivation

`quickSpinEnumerationNode` LogicBlockLUTInputActivation

12.2.1.521 LogicBlockLUTInputSelector

`quickSpinEnumerationNode` LogicBlockLUTInputSelector

12.2.1.522 LogicBlockLUTInputSource

`quickSpinEnumerationNode` LogicBlockLUTInputSource

12.2.1.523 LogicBlockLUTOutputValue

`quickSpinBooleanNode` LogicBlockLUTOutputValue

12.2.1.524 LogicBlockLUTOutputValueAll

`quickSpinIntegerNode` LogicBlockLUTOutputValueAll

12.2.1.525 LogicBlockLUTRowIndex

[quickSpinIntegerNode](#) LogicBlockLUTRowIndex

12.2.1.526 LogicBlockLUTSelector

[quickSpinEnumerationNode](#) LogicBlockLUTSelector

12.2.1.527 LogicBlockSelector

[quickSpinEnumerationNode](#) LogicBlockSelector

12.2.1.528 LUTEnable

[quickSpinBooleanNode](#) LUTEnable

12.2.1.529 LUTIndex

[quickSpinIntegerNode](#) LUTIndex

12.2.1.530 LUTSelector

[quickSpinEnumerationNode](#) LUTSelector

12.2.1.531 LUTValue

[quickSpinIntegerNode](#) LUTValue

12.2.1.532 LUTValueAll

[quickSpinRegisterNode](#) LUTValueAll

12.2.1.533 MaxDeviceResetTime

[quickSpinIntegerNode](#) MaxDeviceResetTime

12.2.1.534 OffsetX

[quickSpinIntegerNode](#) OffsetX

12.2.1.535 OffsetY

[quickSpinIntegerNode](#) OffsetY

12.2.1.536 PacketResendRequestCount

[quickSpinIntegerNode](#) PacketResendRequestCount

12.2.1.537 PayloadSize

[quickSpinIntegerNode](#) PayloadSize

12.2.1.538 PixelColorFilter

[quickSpinEnumerationNode](#) PixelColorFilter

12.2.1.539 PixelDynamicRangeMax

[quickSpinIntegerNode](#) PixelDynamicRangeMax

12.2.1.540 PixelDynamicRangeMin

[quickSpinIntegerNode](#) PixelDynamicRangeMin

12.2.1.541 PixelFormat

[quickSpinEnumerationNode](#) PixelFormat

12.2.1.542 PixelFormatInfoID

[quickSpinIntegerNode](#) PixelFormatInfoID

12.2.1.543 PixelFormatInfoSelector

[quickSpinEnumerationNode](#) PixelFormatInfoSelector

12.2.1.544 PixelSize

[quickSpinEnumerationNode](#) PixelSize

12.2.1.545 PowerSupplyCurrent

[quickSpinFloatNode](#) PowerSupplyCurrent

12.2.1.546 PowerSupplyVoltage

[quickSpinFloatNode](#) PowerSupplyVoltage

12.2.1.547 RegionDestination

[quickSpinEnumerationNode](#) RegionDestination

12.2.1.548 RegionMode

[quickSpinEnumerationNode](#) RegionMode

12.2.1.549 RegionSelector

`quickSpinEnumerationNode` RegionSelector

12.2.1.550 ReverseX

`quickSpinBooleanNode` ReverseX

12.2.1.551 ReverseY

`quickSpinBooleanNode` ReverseY

12.2.1.552 RgbTransformLightSource

`quickSpinEnumerationNode` RgbTransformLightSource

12.2.1.553 Saturation

`quickSpinFloatNode` Saturation

12.2.1.554 SaturationEnable

`quickSpinBooleanNode` SaturationEnable

12.2.1.555 Scan3dAxisMax

`quickSpinFloatNode` Scan3dAxisMax

12.2.1.556 Scan3dAxisMin

`quickSpinFloatNode` Scan3dAxisMin

12.2.1.557 Scan3dCoordinateOffset

`quickSpinFloatNode` `Scan3dCoordinateOffset`

12.2.1.558 Scan3dCoordinateReferenceSelector

`quickSpinEnumerationNode` `Scan3dCoordinateReferenceSelector`

12.2.1.559 Scan3dCoordinateReferenceValue

`quickSpinFloatNode` `Scan3dCoordinateReferenceValue`

12.2.1.560 Scan3dCoordinateScale

`quickSpinFloatNode` `Scan3dCoordinateScale`

12.2.1.561 Scan3dCoordinateSelector

`quickSpinEnumerationNode` `Scan3dCoordinateSelector`

12.2.1.562 Scan3dCoordinateSystem

`quickSpinEnumerationNode` `Scan3dCoordinateSystem`

12.2.1.563 Scan3dCoordinateSystemReference

`quickSpinEnumerationNode` `Scan3dCoordinateSystemReference`

12.2.1.564 Scan3dCoordinateTransformSelector

`quickSpinEnumerationNode` `Scan3dCoordinateTransformSelector`

12.2.1.565 Scan3dDistanceUnit

`quickSpinEnumerationNode` Scan3dDistanceUnit

12.2.1.566 Scan3dInvalidDataFlag

`quickSpinBooleanNode` Scan3dInvalidDataFlag

12.2.1.567 Scan3dInvalidDataValue

`quickSpinFloatNode` Scan3dInvalidDataValue

12.2.1.568 Scan3dOutputMode

`quickSpinEnumerationNode` Scan3dOutputMode

12.2.1.569 Scan3dTransformValue

`quickSpinFloatNode` Scan3dTransformValue

12.2.1.570 SensorDescription

`quickSpinStringNode` SensorDescription

12.2.1.571 SensorDigitizationTaps

`quickSpinEnumerationNode` SensorDigitizationTaps

12.2.1.572 SensorHeight

`quickSpinIntegerNode` SensorHeight

12.2.1.573 SensorShutterMode

`quickSpinEnumerationNode` SensorShutterMode

12.2.1.574 SensorTaps

`quickSpinEnumerationNode` SensorTaps

12.2.1.575 SensorWidth

`quickSpinIntegerNode` SensorWidth

12.2.1.576 SequencerConfigurationMode

`quickSpinEnumerationNode` SequencerConfigurationMode

12.2.1.577 SequencerConfigurationValid

`quickSpinEnumerationNode` SequencerConfigurationValid

12.2.1.578 SequencerFeatureEnable

`quickSpinBooleanNode` SequencerFeatureEnable

12.2.1.579 SequencerMode

`quickSpinEnumerationNode` SequencerMode

12.2.1.580 SequencerPathSelector

`quickSpinIntegerNode` SequencerPathSelector

12.2.1.581 SequencerSetActive

[quickSpinIntegerNode](#) SequencerSetActive

12.2.1.582 SequencerSetLoad

[quickSpinCommandNode](#) SequencerSetLoad

12.2.1.583 SequencerSetNext

[quickSpinIntegerNode](#) SequencerSetNext

12.2.1.584 SequencerSetSave

[quickSpinCommandNode](#) SequencerSetSave

12.2.1.585 SequencerSetSelector

[quickSpinIntegerNode](#) SequencerSetSelector

12.2.1.586 SequencerSetStart

[quickSpinIntegerNode](#) SequencerSetStart

12.2.1.587 SequencerSetValid

[quickSpinEnumerationNode](#) SequencerSetValid

12.2.1.588 SequencerTriggerActivation

[quickSpinEnumerationNode](#) SequencerTriggerActivation

12.2.1.589 SequencerTriggerSource

`quickSpinEnumerationNode` SequencerTriggerSource

12.2.1.590 SerialPortBaudRate

`quickSpinEnumerationNode` SerialPortBaudRate

12.2.1.591 SerialPortDataBits

`quickSpinIntegerNode` SerialPortDataBits

12.2.1.592 SerialPortParity

`quickSpinEnumerationNode` SerialPortParity

12.2.1.593 SerialPortSelector

`quickSpinEnumerationNode` SerialPortSelector

12.2.1.594 SerialPortSource

`quickSpinEnumerationNode` SerialPortSource

12.2.1.595 SerialPortStopBits

`quickSpinEnumerationNode` SerialPortStopBits

12.2.1.596 SerialReceiveFramingErrorCount

`quickSpinIntegerNode` SerialReceiveFramingErrorCount

12.2.1.597 SerialReceiveParityErrorCount

`quickSpinIntegerNode` SerialReceiveParityErrorCount

12.2.1.598 SerialReceiveQueueClear

`quickSpinCommandNode` SerialReceiveQueueClear

12.2.1.599 SerialReceiveQueueCurrentCharacterCount

`quickSpinIntegerNode` SerialReceiveQueueCurrentCharacterCount

12.2.1.600 SerialReceiveQueueMaxCharacterCount

`quickSpinIntegerNode` SerialReceiveQueueMaxCharacterCount

12.2.1.601 SerialTransmitQueueCurrentCharacterCount

`quickSpinIntegerNode` SerialTransmitQueueCurrentCharacterCount

12.2.1.602 SerialTransmitQueueMaxCharacterCount

`quickSpinIntegerNode` SerialTransmitQueueMaxCharacterCount

12.2.1.603 Sharpening

`quickSpinFloatNode` Sharpening

12.2.1.604 SharpeningAuto

`quickSpinBooleanNode` SharpeningAuto

12.2.1.605 SharpeningEnable

`quickSpinBooleanNode` SharpeningEnable

12.2.1.606 SharpeningThreshold

`quickSpinFloatNode` SharpeningThreshold

12.2.1.607 SoftwareSignalPulse

`quickSpinCommandNode` SoftwareSignalPulse

12.2.1.608 SoftwareSignalSelector

`quickSpinEnumerationNode` SoftwareSignalSelector

12.2.1.609 SourceCount

`quickSpinIntegerNode` SourceCount

12.2.1.610 SourceSelector

`quickSpinEnumerationNode` SourceSelector

12.2.1.611 Test0001

`quickSpinIntegerNode` Test0001

12.2.1.612 TestEventGenerate

`quickSpinCommandNode` TestEventGenerate

12.2.1.613 TestPattern

`quickSpinEnumerationNode` TestPattern

12.2.1.614 TestPatternGeneratorSelector

`quickSpinEnumerationNode` TestPatternGeneratorSelector

12.2.1.615 TestPendingAck

`quickSpinIntegerNode` TestPendingAck

12.2.1.616 TimerDelay

`quickSpinFloatNode` TimerDelay

12.2.1.617 TimerDuration

`quickSpinFloatNode` TimerDuration

12.2.1.618 TimerReset

`quickSpinCommandNode` TimerReset

12.2.1.619 TimerSelector

`quickSpinEnumerationNode` TimerSelector

12.2.1.620 TimerStatus

`quickSpinEnumerationNode` TimerStatus

12.2.1.621 TimerTriggerActivation

[quickSpinEnumerationNode](#) TimerTriggerActivation

12.2.1.622 TimerTriggerSource

[quickSpinEnumerationNode](#) TimerTriggerSource

12.2.1.623 TimerValue

[quickSpinFloatNode](#) TimerValue

12.2.1.624 Timestamp

[quickSpinIntegerNode](#) Timestamp

12.2.1.625 TimestampLatch

[quickSpinCommandNode](#) TimestampLatch

12.2.1.626 TimestampLatchValue

[quickSpinIntegerNode](#) TimestampLatchValue

12.2.1.627 TimestampReset

[quickSpinCommandNode](#) TimestampReset

12.2.1.628 TLParamsLocked

[quickSpinIntegerNode](#) TLParamsLocked

12.2.1.629 TransferAbort

[quickSpinCommandNode](#) TransferAbort

12.2.1.630 TransferBlockCount

[quickSpinIntegerNode](#) TransferBlockCount

12.2.1.631 TransferBurstCount

[quickSpinIntegerNode](#) TransferBurstCount

12.2.1.632 TransferComponentSelector

[quickSpinEnumerationNode](#) TransferComponentSelector

12.2.1.633 TransferControlMode

[quickSpinEnumerationNode](#) TransferControlMode

12.2.1.634 TransferOperationMode

[quickSpinEnumerationNode](#) TransferOperationMode

12.2.1.635 TransferPause

[quickSpinCommandNode](#) TransferPause

12.2.1.636 TransferQueueCurrentBlockCount

[quickSpinIntegerNode](#) TransferQueueCurrentBlockCount

12.2.1.637 TransferQueueMaxBlockCount

`quickSpinIntegerNode` TransferQueueMaxBlockCount

12.2.1.638 TransferQueueMode

`quickSpinEnumerationNode` TransferQueueMode

12.2.1.639 TransferQueueOverflowCount

`quickSpinIntegerNode` TransferQueueOverflowCount

12.2.1.640 TransferResume

`quickSpinCommandNode` TransferResume

12.2.1.641 TransferSelector

`quickSpinEnumerationNode` TransferSelector

12.2.1.642 TransferStart

`quickSpinCommandNode` TransferStart

12.2.1.643 TransferStatus

`quickSpinBooleanNode` TransferStatus

12.2.1.644 TransferStatusSelector

`quickSpinEnumerationNode` TransferStatusSelector

12.2.1.645 TransferStop

`quickSpinCommandNode` TransferStop

12.2.1.646 TransferStreamChannel

`quickSpinIntegerNode` TransferStreamChannel

12.2.1.647 TransferTriggerActivation

`quickSpinEnumerationNode` TransferTriggerActivation

12.2.1.648 TransferTriggerMode

`quickSpinEnumerationNode` TransferTriggerMode

12.2.1.649 TransferTriggerSelector

`quickSpinEnumerationNode` TransferTriggerSelector

12.2.1.650 TransferTriggerSource

`quickSpinEnumerationNode` TransferTriggerSource

12.2.1.651 TriggerActivation

`quickSpinEnumerationNode` TriggerActivation

12.2.1.652 TriggerDelay

`quickSpinFloatNode` TriggerDelay

12.2.1.653 TriggerDivider

`quickSpinIntegerNode` TriggerDivider

12.2.1.654 TriggerEventTest

`quickSpinCommandNode` TriggerEventTest

12.2.1.655 TriggerMode

`quickSpinEnumerationNode` TriggerMode

12.2.1.656 TriggerMultiplier

`quickSpinIntegerNode` TriggerMultiplier

12.2.1.657 TriggerOverlap

`quickSpinEnumerationNode` TriggerOverlap

12.2.1.658 TriggerSelector

`quickSpinEnumerationNode` TriggerSelector

12.2.1.659 TriggerSoftware

`quickSpinCommandNode` TriggerSoftware

12.2.1.660 TriggerSource

`quickSpinEnumerationNode` TriggerSource

12.2.1.661 UserOutputSelector

`quickSpinEnumerationNode` UserOutputSelector

12.2.1.662 UserOutputValue

`quickSpinBooleanNode` UserOutputValue

12.2.1.663 UserOutputValueAll

`quickSpinIntegerNode` UserOutputValueAll

12.2.1.664 UserOutputValueAllMask

`quickSpinIntegerNode` UserOutputValueAllMask

12.2.1.665 UserSetDefault

`quickSpinEnumerationNode` UserSetDefault

12.2.1.666 UserSetFeatureEnable

`quickSpinBooleanNode` UserSetFeatureEnable

12.2.1.667 UserSetLoad

`quickSpinCommandNode` UserSetLoad

12.2.1.668 UserSetSave

`quickSpinCommandNode` UserSetSave

12.2.1.669 UserSetSelector

[quickSpinEnumerationNode](#) UserSetSelector

12.2.1.670 V3_3Enable

[quickSpinBooleanNode](#) V3_3Enable

12.2.1.671 WhiteClip

[quickSpinFloatNode](#) WhiteClip

12.2.1.672 WhiteClipSelector

[quickSpinEnumerationNode](#) WhiteClipSelector

12.2.1.673 Width

[quickSpinIntegerNode](#) Width

12.2.1.674 WidthMax

[quickSpinIntegerNode](#) WidthMax

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

- [include/spinc/QuickSpinDefsC.h](#)

12.3 _quickSpinTLDevice Struct Reference

Data Fields

- quickSpinStringNode DeviceID
- quickSpinStringNode DeviceSerialNumber
- quickSpinStringNode DeviceUserID
- quickSpinStringNode DeviceVendorName
- quickSpinStringNode DeviceModelName
- quickSpinStringNode DeviceVersion
- quickSpinIntegerNode DeviceBootloaderVersion
- quickSpinEnumerationNode DeviceType
- quickSpinStringNode DeviceDisplayName
- quickSpinEnumerationNode DeviceAccessStatus
- quickSpinIntegerNode DeviceLinkSpeed
- quickSpinStringNode DeviceDriverVersion
- quickSpinBooleanNode DeviceIsUpdater
- quickSpinEnumerationNode GenICamXMLLocation
- quickSpinStringNode GenICamXMLPath
- quickSpinEnumerationNode GUIXMLLocation
- quickSpinStringNode GUIXMLPath
- quickSpinEnumerationNode GevCCP
- quickSpinIntegerNode GevDeviceMACAddress
- quickSpinIntegerNode GevDeviceIPAddress
- quickSpinIntegerNode GevDeviceSubnetMask
- quickSpinIntegerNode GevDeviceGateway
- quickSpinIntegerNode GevVersionMajor
- quickSpinIntegerNode GevVersionMinor
- quickSpinBooleanNode GevDeviceModelsBigEndian
- quickSpinIntegerNode GevDeviceReadAndWriteTimeout
- quickSpinIntegerNode GevDeviceMaximumRetryCount
- quickSpinIntegerNode GevDevicePort
- quickSpinCommandNode GevDeviceDiscoverMaximumPacketSize
- quickSpinIntegerNode GevDeviceMaximumPacketSize
- quickSpinBooleanNode GevDeviceIsWrongSubnet
- quickSpinCommandNode GevDeviceAutoForceIP
- quickSpinCommandNode GevDeviceForceIP
- quickSpinIntegerNode GevDeviceForceIPAddress
- quickSpinIntegerNode GevDeviceForceSubnetMask
- quickSpinIntegerNode GevDeviceForceGateway
- quickSpinBooleanNode DeviceMulticastMonitorMode
- quickSpinEnumerationNode DeviceEndiannessMechanism
- quickSpinCommandNode DeviceReset
- quickSpinStringNode DeviceInstanceId
- quickSpinStringNode DeviceLocation
- quickSpinEnumerationNode DeviceCurrentSpeed
- quickSpinBooleanNode DeviceU3VProtocol
- quickSpinStringNode DevicePortId

12.3.1 Field Documentation

12.3.1.1 DeviceAccessStatus

[quickSpinEnumerationNode](#) DeviceAccessStatus

12.3.1.2 DeviceBootloaderVersion

[quickSpinIntegerNode](#) DeviceBootloaderVersion

12.3.1.3 DeviceCurrentSpeed

[quickSpinEnumerationNode](#) DeviceCurrentSpeed

12.3.1.4 DeviceDisplayName

[quickSpinStringNode](#) DeviceDisplayName

12.3.1.5 DeviceDriverVersion

[quickSpinStringNode](#) DeviceDriverVersion

12.3.1.6 DeviceEndiannessMechanism

[quickSpinEnumerationNode](#) DeviceEndiannessMechanism

12.3.1.7 DeviceID

[quickSpinStringNode](#) DeviceID

12.3.1.8 DeviceInstanceId

[quickSpinStringNode](#) DeviceInstanceId

12.3.1.9 DeviceIsUpdater

`quickSpinBooleanNode` DeviceIsUpdater

12.3.1.10 DeviceLinkSpeed

`quickSpinIntegerNode` DeviceLinkSpeed

12.3.1.11 DeviceLocation

`quickSpinStringNode` DeviceLocation

12.3.1.12 DeviceModelName

`quickSpinStringNode` DeviceModelName

12.3.1.13 DeviceMulticastMonitorMode

`quickSpinBooleanNode` DeviceMulticastMonitorMode

12.3.1.14 DevicePortId

`quickSpinStringNode` DevicePortId

12.3.1.15 DeviceReset

`quickSpinCommandNode` DeviceReset

12.3.1.16 DeviceSerialNumber

`quickSpinStringNode` DeviceSerialNumber

12.3.1.17 DeviceType

`quickSpinEnumerationNode` DeviceType

12.3.1.18 DeviceU3VProtocol

`quickSpinBooleanNode` DeviceU3VProtocol

12.3.1.19 DeviceUserID

`quickSpinStringNode` DeviceUserID

12.3.1.20 DeviceVendorName

`quickSpinStringNode` DeviceVendorName

12.3.1.21 DeviceVersion

`quickSpinStringNode` DeviceVersion

12.3.1.22 GenICamXMLLocation

`quickSpinEnumerationNode` GenICamXMLLocation

12.3.1.23 GenICamXMLPath

`quickSpinStringNode` GenICamXMLPath

12.3.1.24 GevCCP

`quickSpinEnumerationNode` GevCCP

12.3.1.25 **GevDeviceAutoForceIP**

[quickSpinCommandNode](#) [GevDeviceAutoForceIP](#)

12.3.1.26 **GevDeviceDiscoverMaximumPacketSize**

[quickSpinCommandNode](#) [GevDeviceDiscoverMaximumPacketSize](#)

12.3.1.27 **GevDeviceForceGateway**

[quickSpinIntegerNode](#) [GevDeviceForceGateway](#)

12.3.1.28 **GevDeviceForceIP**

[quickSpinCommandNode](#) [GevDeviceForceIP](#)

12.3.1.29 **GevDeviceForceIPAddress**

[quickSpinIntegerNode](#) [GevDeviceForceIPAddress](#)

12.3.1.30 **GevDeviceForceSubnetMask**

[quickSpinIntegerNode](#) [GevDeviceForceSubnetMask](#)

12.3.1.31 **GevDeviceGateway**

[quickSpinIntegerNode](#) [GevDeviceGateway](#)

12.3.1.32 **GevDeviceIPAddress**

[quickSpinIntegerNode](#) [GevDeviceIPAddress](#)

12.3.1.33 **GevDeviceIsWrongSubnet**

`quickSpinBooleanNode` `GevDeviceIsWrongSubnet`

12.3.1.34 **GevDeviceMACAddress**

`quickSpinIntegerNode` `GevDeviceMACAddress`

12.3.1.35 **GevDeviceMaximumPacketSize**

`quickSpinIntegerNode` `GevDeviceMaximumPacketSize`

12.3.1.36 **GevDeviceMaximumRetryCount**

`quickSpinIntegerNode` `GevDeviceMaximumRetryCount`

12.3.1.37 **GevDeviceModelsBigEndian**

`quickSpinBooleanNode` `GevDeviceModeIsBigEndian`

12.3.1.38 **GevDevicePort**

`quickSpinIntegerNode` `GevDevicePort`

12.3.1.39 **GevDeviceReadAndWriteTimeout**

`quickSpinIntegerNode` `GevDeviceReadAndWriteTimeout`

12.3.1.40 **GevDeviceSubnetMask**

`quickSpinIntegerNode` `GevDeviceSubnetMask`

12.3.1.41 **GevVersionMajor**

[quickSpinIntegerNode](#) [GevVersionMajor](#)

12.3.1.42 **GevVersionMinor**

[quickSpinIntegerNode](#) [GevVersionMinor](#)

12.3.1.43 **GUIXMLLocation**

[quickSpinEnumerationNode](#) [GUIXMLLocation](#)

12.3.1.44 **GUIXMLPath**

[quickSpinStringNode](#) [GUIXMLPath](#)

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

- [include/spinc/TransportLayerDeviceC.h](#)

12.4 **_quickSpinTLInterface Struct Reference**

Data Fields

- [quickSpinStringNode](#) [InterfaceID](#)
- [quickSpinStringNode](#) [InterfaceDisplayName](#)
- [quickSpinEnumerationNode](#) [InterfaceType](#)
- [quickSpinIntegerNode](#) [GevInterfaceGatewaySelector](#)
- [quickSpinIntegerNode](#) [GevInterfaceGateway](#)
- [quickSpinIntegerNode](#) [GevInterfaceMACAddress](#)
- [quickSpinIntegerNode](#) [GevInterfaceSubnetSelector](#)
- [quickSpinIntegerNode](#) [GevInterfaceSubnetIPAddress](#)
- [quickSpinIntegerNode](#) [GevInterfaceSubnetMask](#)
- [quickSpinIntegerNode](#) [GevInterfaceTransmitLinkSpeed](#)
- [quickSpinIntegerNode](#) [GevInterfaceReceiveLinkSpeed](#)
- [quickSpinIntegerNode](#) [GevInterfaceMTU](#)
- [quickSpinEnumerationNode](#) [POEStatus](#)
- [quickSpinEnumerationNode](#) [FilterDriverStatus](#)
- [quickSpinIntegerNode](#) [GevActionDeviceKey](#)
- [quickSpinIntegerNode](#) [GevActionGroupKey](#)
- [quickSpinIntegerNode](#) [GevActionGroupMask](#)
- [quickSpinIntegerNode](#) [GevActionTime](#)

- [quickSpinCommandNode](#) ActionCommand
- [quickSpinStringNode](#) DeviceUnlock
- [quickSpinCommandNode](#) DeviceUpdateList
- [quickSpinIntegerNode](#) DeviceCount
- [quickSpinIntegerNode](#) DeviceSelector
- [quickSpinStringNode](#) DeviceID
- [quickSpinStringNode](#) DeviceVendorName
- [quickSpinStringNode](#) DeviceModelName
- [quickSpinStringNode](#) DeviceSerialNumber
- [quickSpinEnumerationNode](#) DeviceAccessStatus
- [quickSpinIntegerNode](#) GevDeviceIPAddress
- [quickSpinIntegerNode](#) GevDeviceSubnetMask
- [quickSpinIntegerNode](#) GevDeviceGateway
- [quickSpinIntegerNode](#) GevDeviceMACAddress
- [quickSpinIntegerNode](#) IncompatibleDeviceCount
- [quickSpinIntegerNode](#) IncompatibleDeviceSelector
- [quickSpinStringNode](#) IncompatibleDeviceID
- [quickSpinStringNode](#) IncompatibleDeviceVendorName
- [quickSpinStringNode](#) IncompatibleDeviceModelName
- [quickSpinIntegerNode](#) IncompatibleGevDeviceIPAddress
- [quickSpinIntegerNode](#) IncompatibleGevDeviceSubnetMask
- [quickSpinIntegerNode](#) IncompatibleGevDeviceMACAddress
- [quickSpinCommandNode](#) GevDeviceForceIP
- [quickSpinIntegerNode](#) GevDeviceForceIPAddress
- [quickSpinIntegerNode](#) GevDeviceForceSubnetMask
- [quickSpinIntegerNode](#) GevDeviceForceGateway
- [quickSpinCommandNode](#) GevDeviceAutoForceIP
- [quickSpinStringNode](#) HostAdapterName
- [quickSpinStringNode](#) HostAdapterVendor
- [quickSpinStringNode](#) HostAdapterDriverVersion

12.4.1 Field Documentation

12.4.1.1 ActionCommand

[quickSpinCommandNode](#) ActionCommand

12.4.1.2 DeviceAccessStatus

[quickSpinEnumerationNode](#) DeviceAccessStatus

12.4.1.3 DeviceCount

`quickSpinIntegerNode` DeviceCount

12.4.1.4 DeviceID

`quickSpinStringNode` DeviceID

12.4.1.5 DeviceModelName

`quickSpinStringNode` DeviceModelName

12.4.1.6 DeviceSelector

`quickSpinIntegerNode` DeviceSelector

12.4.1.7 DeviceSerialNumber

`quickSpinStringNode` DeviceSerialNumber

12.4.1.8 DeviceUnlock

`quickSpinStringNode` DeviceUnlock

12.4.1.9 DeviceUpdateList

`quickSpinCommandNode` DeviceUpdateList

12.4.1.10 DeviceVendorName

`quickSpinStringNode` DeviceVendorName

12.4.1.11 FilterDriverStatus

`quickSpinEnumerationNode` FilterDriverStatus

12.4.1.12 GevActionDeviceKey

`quickSpinIntegerNode` GevActionDeviceKey

12.4.1.13 GevActionGroupKey

`quickSpinIntegerNode` GevActionGroupKey

12.4.1.14 GevActionGroupMask

`quickSpinIntegerNode` GevActionGroupMask

12.4.1.15 GevActionTime

`quickSpinIntegerNode` GevActionTime

12.4.1.16 GevDeviceAutoForceIP

`quickSpinCommandNode` GevDeviceAutoForceIP

12.4.1.17 GevDeviceForceGateway

`quickSpinIntegerNode` GevDeviceForceGateway

12.4.1.18 GevDeviceForceIP

`quickSpinCommandNode` GevDeviceForceIP

12.4.1.19 **GevDeviceForceIPAddress**

[quickSpinIntegerNode](#) [GevDeviceForceIPAddress](#)

12.4.1.20 **GevDeviceForceSubnetMask**

[quickSpinIntegerNode](#) [GevDeviceForceSubnetMask](#)

12.4.1.21 **GevDeviceGateway**

[quickSpinIntegerNode](#) [GevDeviceGateway](#)

12.4.1.22 **GevDeviceIPAddress**

[quickSpinIntegerNode](#) [GevDeviceIPAddress](#)

12.4.1.23 **GevDeviceMACAddress**

[quickSpinIntegerNode](#) [GevDeviceMACAddress](#)

12.4.1.24 **GevDeviceSubnetMask**

[quickSpinIntegerNode](#) [GevDeviceSubnetMask](#)

12.4.1.25 **GevInterfaceGateway**

[quickSpinIntegerNode](#) [GevInterfaceGateway](#)

12.4.1.26 **GevInterfaceGatewaySelector**

[quickSpinIntegerNode](#) [GevInterfaceGatewaySelector](#)

12.4.1.27 **GevInterfaceMACAddress**

`quickSpinIntegerNode` `GevInterfaceMACAddress`

12.4.1.28 **GevInterfaceMTU**

`quickSpinIntegerNode` `GevInterfaceMTU`

12.4.1.29 **GevInterfaceReceiveLinkSpeed**

`quickSpinIntegerNode` `GevInterfaceReceiveLinkSpeed`

12.4.1.30 **GevInterfaceSubnetIPAddress**

`quickSpinIntegerNode` `GevInterfaceSubnetIPAddress`

12.4.1.31 **GevInterfaceSubnetMask**

`quickSpinIntegerNode` `GevInterfaceSubnetMask`

12.4.1.32 **GevInterfaceSubnetSelector**

`quickSpinIntegerNode` `GevInterfaceSubnetSelector`

12.4.1.33 **GevInterfaceTransmitLinkSpeed**

`quickSpinIntegerNode` `GevInterfaceTransmitLinkSpeed`

12.4.1.34 **HostAdapterDriverVersion**

`quickSpinStringNode` `HostAdapterDriverVersion`

12.4.1.35 HostAdapterName

`quickSpinStringNode` HostAdapterName

12.4.1.36 HostAdapterVendor

`quickSpinStringNode` HostAdapterVendor

12.4.1.37 IncompatibleDeviceCount

`quickSpinIntegerNode` IncompatibleDeviceCount

12.4.1.38 IncompatibleDeviceID

`quickSpinStringNode` IncompatibleDeviceID

12.4.1.39 IncompatibleDeviceModelName

`quickSpinStringNode` IncompatibleDeviceModelName

12.4.1.40 IncompatibleDeviceSelector

`quickSpinIntegerNode` IncompatibleDeviceSelector

12.4.1.41 IncompatibleDeviceVendorName

`quickSpinStringNode` IncompatibleDeviceVendorName

12.4.1.42 IncompatibleGevDeviceIPAddress

`quickSpinIntegerNode` IncompatibleGevDeviceIPAddress

12.4.1.43 IncompatibleGevDeviceMACAddress

[quickSpinIntegerNode](#) IncompatibleGevDeviceMACAddress

12.4.1.44 IncompatibleGevDeviceSubnetMask

[quickSpinIntegerNode](#) IncompatibleGevDeviceSubnetMask

12.4.1.45 InterfaceDisplayName

[quickSpinStringNode](#) InterfaceDisplayName

12.4.1.46 InterfaceID

[quickSpinStringNode](#) InterfaceID

12.4.1.47 InterfaceType

[quickSpinEnumerationNode](#) InterfaceType

12.4.1.48 POEStatus

[quickSpinEnumerationNode](#) POEStatus

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

- [include/spinc/TransportLayerInterfaceC.h](#)

12.5 _quickSpinTLStream Struct Reference

Data Fields

- [quickSpinStringNode](#) StreamID
- [quickSpinEnumerationNode](#) StreamType
- [quickSpinEnumerationNode](#) StreamMode
- [quickSpinIntegerNode](#) StreamBufferCountManual
- [quickSpinIntegerNode](#) StreamBufferCountResult
- [quickSpinIntegerNode](#) StreamBufferCountMax
- [quickSpinEnumerationNode](#) StreamBufferCountMode
- [quickSpinEnumerationNode](#) StreamBufferHandlingMode
- [quickSpinIntegerNode](#) StreamAnnounceBufferMinimum
- [quickSpinIntegerNode](#) StreamAnnouncedBufferCount
- [quickSpinIntegerNode](#) StreamStartedFrameCount
- [quickSpinIntegerNode](#) StreamDeliveredFrameCount
- [quickSpinIntegerNode](#) StreamReceivedFrameCount
- [quickSpinIntegerNode](#) StreamIncompleteFrameCount
- [quickSpinIntegerNode](#) StreamLostFrameCount
- [quickSpinIntegerNode](#) StreamDroppedFrameCount
- [quickSpinIntegerNode](#) StreamInputBufferCount
- [quickSpinIntegerNode](#) StreamOutputBufferCount
- [quickSpinBooleanNode](#) StreamIsGrabbing
- [quickSpinIntegerNode](#) StreamChunkCountMaximum
- [quickSpinIntegerNode](#) StreamBufferAlignment
- [quickSpinBooleanNode](#) StreamCRCCheckEnable
- [quickSpinIntegerNode](#) StreamReceivedPacketCount
- [quickSpinIntegerNode](#) StreamMissedPacketCount
- [quickSpinBooleanNode](#) StreamPacketResendEnable
- [quickSpinIntegerNode](#) StreamPacketResendTimeout
- [quickSpinIntegerNode](#) StreamPacketResendMaxRequests
- [quickSpinIntegerNode](#) StreamPacketResendRequestCount
- [quickSpinIntegerNode](#) StreamPacketResendRequestSuccessCount
- [quickSpinIntegerNode](#) StreamPacketResendRequestedPacketCount
- [quickSpinIntegerNode](#) StreamPacketResendReceivedPacketCount
- [quickSpinIntegerNode](#) StreamBlockTransferSize

12.5.1 Field Documentation

12.5.1.1 StreamAnnounceBufferMinimum

[quickSpinIntegerNode](#) StreamAnnounceBufferMinimum

12.5.1.2 StreamAnnouncedBufferCount

[quickSpinIntegerNode](#) StreamAnnouncedBufferCount

12.5.1.3 StreamBlockTransferSize

[quickSpinIntegerNode](#) StreamBlockTransferSize

12.5.1.4 StreamBufferAlignment

[quickSpinIntegerNode](#) StreamBufferAlignment

12.5.1.5 StreamBufferCountManual

[quickSpinIntegerNode](#) StreamBufferCountManual

12.5.1.6 StreamBufferCountMax

[quickSpinIntegerNode](#) StreamBufferCountMax

12.5.1.7 StreamBufferCountMode

[quickSpinEnumerationNode](#) StreamBufferCountMode

12.5.1.8 StreamBufferCountResult

[quickSpinIntegerNode](#) StreamBufferCountResult

12.5.1.9 StreamBufferHandlingMode

[quickSpinEnumerationNode](#) StreamBufferHandlingMode

12.5.1.10 StreamChunkCountMaximum

[quickSpinIntegerNode](#) StreamChunkCountMaximum

12.5.1.11 StreamCRCCheckEnable

`quickSpinBooleanNode` StreamCRCCheckEnable

12.5.1.12 StreamDeliveredFrameCount

`quickSpinIntegerNode` StreamDeliveredFrameCount

12.5.1.13 StreamDroppedFrameCount

`quickSpinIntegerNode` StreamDroppedFrameCount

12.5.1.14 StreamID

`quickSpinStringNode` StreamID

12.5.1.15 StreamIncompleteFrameCount

`quickSpinIntegerNode` StreamIncompleteFrameCount

12.5.1.16 StreamInputBufferCount

`quickSpinIntegerNode` StreamInputBufferCount

12.5.1.17 StreamIsGrabbing

`quickSpinBooleanNode` StreamIsGrabbing

12.5.1.18 StreamLostFrameCount

`quickSpinIntegerNode` StreamLostFrameCount

12.5.1.19 StreamMissedPacketCount

[quickSpinIntegerNode](#) StreamMissedPacketCount

12.5.1.20 StreamMode

[quickSpinEnumerationNode](#) StreamMode

12.5.1.21 StreamOutputBufferCount

[quickSpinIntegerNode](#) StreamOutputBufferCount

12.5.1.22 StreamPacketResendEnable

[quickSpinBooleanNode](#) StreamPacketResendEnable

12.5.1.23 StreamPacketResendMaxRequests

[quickSpinIntegerNode](#) StreamPacketResendMaxRequests

12.5.1.24 StreamPacketResendReceivedPacketCount

[quickSpinIntegerNode](#) StreamPacketResendReceivedPacketCount

12.5.1.25 StreamPacketResendRequestCount

[quickSpinIntegerNode](#) StreamPacketResendRequestCount

12.5.1.26 StreamPacketResendRequestedPacketCount

[quickSpinIntegerNode](#) StreamPacketResendRequestedPacketCount

12.5.1.27 StreamPacketResendRequestSuccessCount

[quickSpinIntegerNode](#) StreamPacketResendRequestSuccessCount

12.5.1.28 StreamPacketResendTimeout

[quickSpinIntegerNode](#) StreamPacketResendTimeout

12.5.1.29 StreamReceivedFrameCount

[quickSpinIntegerNode](#) StreamReceivedFrameCount

12.5.1.30 StreamReceivedPacketCount

[quickSpinIntegerNode](#) StreamReceivedPacketCount

12.5.1.31 StreamStartedFrameCount

[quickSpinIntegerNode](#) StreamStartedFrameCount

12.5.1.32 StreamType

[quickSpinEnumerationNode](#) StreamType

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

- [include/spinc/TransportLayerStreamC.h](#)

12.6 `_quickSpinTLSystem` Struct Reference

Data Fields

- `quickSpinStringNode` TLID
- `quickSpinStringNode` TLVendorName
- `quickSpinStringNode` TLModelName
- `quickSpinStringNode` TLVersion
- `quickSpinStringNode` TLFileName
- `quickSpinStringNode` TLDisplayName
- `quickSpinStringNode` TLPath
- `quickSpinEnumerationNode` TLType
- `quickSpinIntegerNode` GenTLVersionMajor
- `quickSpinIntegerNode` GenTLVersionMinor
- `quickSpinIntegerNode` GenTLSFNCVersionMajor
- `quickSpinIntegerNode` GenTLSFNCVersionMinor
- `quickSpinIntegerNode` GenTLSFNCVersionSubMinor
- `quickSpinIntegerNode` GevVersionMajor
- `quickSpinIntegerNode` GevVersionMinor
- `quickSpinCommandNode` InterfaceUpdateList
- `quickSpinIntegerNode` InterfaceSelector
- `quickSpinStringNode` InterfaceID
- `quickSpinStringNode` InterfaceDisplayName
- `quickSpinIntegerNode` GevInterfaceMACAddress
- `quickSpinIntegerNode` GevInterfaceDefaultIPAddress
- `quickSpinIntegerNode` GevInterfaceDefaultSubnetMask
- `quickSpinIntegerNode` GevInterfaceDefaultGateway
- `quickSpinBooleanNode` EnumerateGEVInterfaces
- `quickSpinBooleanNode` EnumerateUSBInterfaces
- `quickSpinBooleanNode` EnumerateGen2Cameras

12.6.1 Field Documentation

12.6.1.1 EnumerateGen2Cameras

`quickSpinBooleanNode` EnumerateGen2Cameras

12.6.1.2 EnumerateGEVInterfaces

`quickSpinBooleanNode` EnumerateGEVInterfaces

12.6.1.3 EnumerateUSBInterfaces

`quickSpinBooleanNode` EnumerateUSBInterfaces

12.6.1.4 GenTLFNCVersionMajor

`quickSpinIntegerNode` GenTLFNCVersionMajor

12.6.1.5 GenTLFNCVersionMinor

`quickSpinIntegerNode` GenTLFNCVersionMinor

12.6.1.6 GenTLFNCVersionSubMinor

`quickSpinIntegerNode` GenTLFNCVersionSubMinor

12.6.1.7 GenTLVersionMajor

`quickSpinIntegerNode` GenTLVersionMajor

12.6.1.8 GenTLVersionMinor

`quickSpinIntegerNode` GenTLVersionMinor

12.6.1.9 GevInterfaceDefaultGateway

`quickSpinIntegerNode` GevInterfaceDefaultGateway

12.6.1.10 GevInterfaceDefaultIPAddress

`quickSpinIntegerNode` GevInterfaceDefaultIPAddress

12.6.1.11 **GevInterfaceDefaultSubnetMask**

`quickSpinIntegerNode` `GevInterfaceDefaultSubnetMask`

12.6.1.12 **GevInterfaceMACAddress**

`quickSpinIntegerNode` `GevInterfaceMACAddress`

12.6.1.13 **GevVersionMajor**

`quickSpinIntegerNode` `GevVersionMajor`

12.6.1.14 **GevVersionMinor**

`quickSpinIntegerNode` `GevVersionMinor`

12.6.1.15 **InterfaceDisplayName**

`quickSpinStringNode` `InterfaceDisplayName`

12.6.1.16 **InterfaceID**

`quickSpinStringNode` `InterfaceID`

12.6.1.17 **InterfaceSelector**

`quickSpinIntegerNode` `InterfaceSelector`

12.6.1.18 **InterfaceUpdateList**

`quickSpinCommandNode` `InterfaceUpdateList`

12.6.1.19 TLDisplayName

`quickSpinStringNode` TLDisplayName

12.6.1.20 TLFileName

`quickSpinStringNode` TLFileName

12.6.1.21 TLID

`quickSpinStringNode` TLID

12.6.1.22 TLModelName

`quickSpinStringNode` TLModelName

12.6.1.23 TLPath

`quickSpinStringNode` TLPath

12.6.1.24 TLType

`quickSpinEnumerationNode` TLType

12.6.1.25 TLVendorName

`quickSpinStringNode` TLVendorName

12.6.1.26 TLVersion

`quickSpinStringNode` TLVersion

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

- include/spinc/[TransportLayerSystemC.h](#)

12.7 _spinAVIOption Struct Reference

Options for saving uncompressed videos.

Data Fields

- float [frameRate](#)
Frame rate of the stream.
- unsigned int [width](#)
Width of source image.
- unsigned int [height](#)
Height of source image.
- unsigned int [reserved](#) [192]

12.7.1 Detailed Description

Options for saving uncompressed videos.

Used in saving AVI videos with a call to `spinAVIRecorderOpenUncompressed()`.

12.7.2 Field Documentation

12.7.2.1 frameRate

`float frameRate`

Frame rate of the stream.

12.7.2.2 height

`unsigned int height`

Height of source image.

12.7.2.3 reserved

```
unsigned int reserved[192]
```

12.7.2.4 width

```
unsigned int width
```

Width of source image.

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

- include/spinc/[SpinnakerDefsC.h](#)

12.8 _spinBMPOption Struct Reference

Options for saving BMP images.

Data Fields

- [bool8_t indexedColor_8bit](#)
- unsigned int [reserved](#) [16]

Reserved for future use.

12.8.1 Detailed Description

Options for saving BMP images.

Used in saving PPM images with a call to [spinImageSaveBmp\(\)](#).

12.8.2 Field Documentation

12.8.2.1 indexedColor_8bit

```
bool8_t indexedColor_8bit
```

12.8.2.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/spinc/SpinnakerDefsC.h](#)

12.9 _spinChunkData Struct Reference

The type of information that can be obtained from image chunk data.

Data Fields

- double [m_blackLevel](#)
- int64_t [m_frameID](#)
- double [m_exposureTime](#)
- int64_t [m_compressionMode](#)
- double [m_compressionRatio](#)
- int64_t [m_timestamp](#)
- int64_t [m_exposureEndLineStatusAll](#)
- int64_t [m_width](#)
- int64_t [m_image](#)
- int64_t [m_height](#)
- double [m_gain](#)
- int64_t [m_sequencerSetActive](#)
- int64_t [m_crc](#)
- int64_t [m_offsetX](#)
- int64_t [m_offsetY](#)
- int64_t [m_serialDataLength](#)
- int64_t [m_partSelector](#)
- int64_t [m_pixelDynamicRangeMin](#)
- int64_t [m_pixelDynamicRangeMax](#)
- int64_t [m_timestampLatchValue](#)
- int64_t [m_lineStatusAll](#)
- int64_t [m_counterValue](#)
- double [m_timerValue](#)
- int64_t [m_scanLineSelector](#)
- int64_t [m_encoderValue](#)
- int64_t [m_linePitch](#)
- int64_t [m_transferBlockID](#)
- int64_t [m_transferQueueCurrentBlockCount](#)
- int64_t [m_streamChannelID](#)
- double [m_scan3dCoordinateScale](#)
- double [m_scan3dCoordinateOffset](#)
- double [m_scan3dInvalidDataValue](#)
- double [m_scan3dAxisMin](#)
- double [m_scan3dAxisMax](#)
- double [m_scan3dTransformValue](#)
- double [m_scan3dCoordinateReferenceValue](#)
- int64_t [m_inferenceFrameId](#)
- int64_t [m_inferenceResult](#)
- double [m_inferenceConfidence](#)

12.9.1 Detailed Description

The type of information that can be obtained from image chunk data.

12.9.2 Field Documentation

12.9.2.1 m_blackLevel

```
double m_blackLevel
```

12.9.2.2 m_compressionMode

```
int64_t m_compressionMode
```

12.9.2.3 m_compressionRatio

```
double m_compressionRatio
```

12.9.2.4 m_counterValue

```
int64_t m_counterValue
```

12.9.2.5 m_cRC

```
int64_t m_cRC
```

12.9.2.6 m_encoderValue

```
int64_t m_encoderValue
```

12.9.2.7 m_exposureEndLineStatusAll

```
int64_t m_exposureEndLineStatusAll
```

12.9.2.8 m_exposureTime

```
double m_exposureTime
```

12.9.2.9 m_frameID

```
int64_t m_frameID
```

12.9.2.10 m_gain

```
double m_gain
```

12.9.2.11 m_height

```
int64_t m_height
```

12.9.2.12 m_image

```
int64_t m_image
```

12.9.2.13 m_inferenceConfidence

```
double m_inferenceConfidence
```

12.9.2.14 m_inferenceFrameId

```
int64_t m_inferenceFrameId
```

12.9.2.15 m_inferenceResult

```
int64_t m_inferenceResult
```

12.9.2.16 m_linePitch

```
int64_t m_linePitch
```

12.9.2.17 m_lineStatusAll

```
int64_t m_lineStatusAll
```

12.9.2.18 m_offsetX

```
int64_t m_offsetX
```

12.9.2.19 m_offsetY

```
int64_t m_offsetY
```

12.9.2.20 m_partSelector

```
int64_t m_partSelector
```

12.9.2.21 m_pixelDynamicRangeMax

```
int64_t m_pixelDynamicRangeMax
```

12.9.2.22 m_pixelDynamicRangeMin

```
int64_t m_pixelDynamicRangeMin
```

12.9.2.23 m_scan3dAxisMax

```
double m_scan3dAxisMax
```

12.9.2.24 m_scan3dAxisMin

```
double m_scan3dAxisMin
```

12.9.2.25 m_scan3dCoordinateOffset

```
double m_scan3dCoordinateOffset
```

12.9.2.26 m_scan3dCoordinateReferenceValue

```
double m_scan3dCoordinateReferenceValue
```

12.9.2.27 m_scan3dCoordinateScale

```
double m_scan3dCoordinateScale
```

12.9.2.28 m_scan3dInvalidDataValue

```
double m_scan3dInvalidDataValue
```

12.9.2.29 m_scan3dTransformValue

```
double m_scan3dTransformValue
```

12.9.2.30 m_scanLineSelector

```
int64_t m_scanLineSelector
```

12.9.2.31 m_sequencerSetActive

```
int64_t m_sequencerSetActive
```

12.9.2.32 m_serialDataLength

```
int64_t m_serialDataLength
```

12.9.2.33 m_streamChannelID

```
int64_t m_streamChannelID
```

12.9.2.34 m_timerValue

```
double m_timerValue
```

12.9.2.35 m_timestamp

```
int64_t m_timestamp
```

12.9.2.36 m_timestampLatchValue

```
int64_t m_timestampLatchValue
```

12.9.2.37 m_transferBlockID

```
int64_t m_transferBlockID
```

12.9.2.38 m_transferQueueCurrentBlockCount

```
int64_t m_transferQueueCurrentBlockCount
```

12.9.2.39 m_width

```
int64_t m_width
```

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

- include/spinc/[ChunkDataDefC.h](#)

12.10 _spinH264Option Struct Reference

Options for saving H264 videos.

Data Fields

- float [frameRate](#)
Frame rate of the stream.
- unsigned int [width](#)
Width of source image.
- unsigned int [height](#)
Height of source image.
- unsigned int [bitrate](#)
Bitrate to encode at.
- unsigned int [reserved](#) [256]
Reserved for future use.

12.10.1 Detailed Description

Options for saving H264 videos.

Used in saving H264 videos with a call to `spinAVIRecorderOpenH264()`.

12.10.2 Field Documentation

12.10.2.1 bitrate

```
unsigned int bitrate
```

Bitrate to encode at.

12.10.2.2 frameRate

```
float frameRate
```

Frame rate of the stream.

12.10.2.3 height

```
unsigned int height
```

Height of source image.

12.10.2.4 reserved

```
unsigned int reserved[256]
```

Reserved for future use.

12.10.2.5 width

```
unsigned int width
```

Width of source image.

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

- [include/spinc/SpinnakerDefsC.h](#)

12.11 _spinJPEGOption Struct Reference

Options for saving JPEG images.

Data Fields

- [bool8_t 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.

12.11.1 Detailed Description

Options for saving JPEG images.

Used in saving PPM images with a call to [spinImageSaveJpeg\(\)](#).

12.11.2 Field Documentation

12.11.2.1 progressive

```
bool8_t progressive
```

Whether to save as a progressive JPEG file.

12.11.2.2 quality

```
unsigned int quality
```

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

12.11.2.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/spinc/[SpinnakerDefsC.h](#)

12.12 _spinJPG2Option Struct Reference

Options for saving JPEG 2000 images.

Data Fields

- unsigned int [quality](#)
JPEG saving quality in range (1-512).
- unsigned int [reserved](#) [16]
Reserved for future use.

12.12.1 Detailed Description

Options for saving JPEG 2000 images.

Used in saving PPM images with a call to [spinImageSaveJpg2\(\)](#).

12.12.2 Field Documentation

12.12.2.1 [quality](#)

unsigned int [quality](#)

JPEG saving quality in range (1-512).

12.12.2.2 [reserved](#)

unsigned int [reserved](#)[16]

Reserved for future use.

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

- include/spinc/[SpinnakerDefsC.h](#)

12.13 _spinLibraryVersion Struct Reference

Provides easier access to the current version of Spinnaker.

Data Fields

- 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.

12.13.1 Detailed Description

Provides easier access to the current version of Spinnaker.

12.13.2 Field Documentation

12.13.2.1 build

```
unsigned int build
```

Build number of the library.

12.13.2.2 major

```
unsigned int major
```

Major version of the library.

12.13.2.3 minor

```
unsigned int minor
```

Minor version of the library.

12.13.2.4 type

```
unsigned int type
```

Version type of the library.

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

- [include/spinc/SpinnakerDefsC.h](#)

12.14 `_spinMJPGOption` Struct Reference

Options for saving MJPG videos.

Data Fields

- float `frameRate`
Frame rate of the stream.
- unsigned int `quality`
Image quality (1-100)
- unsigned int `width`
Width of source image.
- unsigned int `height`
Height of source image.
- unsigned int `reserved` [192]

12.14.1 Detailed Description

Options for saving MJPG videos.

Used in saving MJPG videos with a call to `spinAVIRecorderOpenMJPEG()`.

12.14.2 Field Documentation

12.14.2.1 `frameRate`

```
float frameRate
```

Frame rate of the stream.

12.14.2.2 `height`

```
unsigned int height
```

Height of source image.

12.14.2.3 `quality`

```
unsigned int quality
```

Image quality (1-100)

12.14.2.4 reserved

```
unsigned int reserved[192]
```

12.14.2.5 width

```
unsigned int width
```

Width of source image.

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

- include/spinc/[SpinnakerDefsC.h](#)

12.15 `_spinPGMOption` Struct Reference

Options for saving PGM images.

Data Fields

- [bool8_t](#) `binaryFile`
Whether to save the PPM as a binary file.
- unsigned int [reserved](#) [16]
Reserved for future use.

12.15.1 Detailed Description

Options for saving PGM images.

12.15.2 Field Documentation

12.15.2.1 `binaryFile`

```
bool8_t binaryFile
```

Whether to save the PPM as a binary file.

12.15.2.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/spinc/[SpinnakerDefsC.h](#)

12.16 _spinPNGOption Struct Reference

Options for saving PNG images.

Data Fields

- [bool8_t interlaced](#)
Whether to save the PNG as interlaced.
- unsigned int [compressionLevel](#)
Compression level (0-9).
- unsigned int [reserved](#) [16]
Reserved for future use.

12.16.1 Detailed Description

Options for saving PNG images.

Used in saving PNG images with a call to [spinImageSavePng\(\)](#).

12.16.2 Field Documentation

12.16.2.1 compressionLevel

```
unsigned int compressionLevel
```

Compression level (0-9).

0 is no compression, 9 is best compression.

12.16.2.2 interlaced

```
bool8_t interlaced
```

Whether to save the PNG as interlaced.

12.16.2.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/spinc/[SpinnakerDefsC.h](#)

12.17 `_spinPPMOption` Struct Reference

Options for saving PPM images.

Data Fields

- `bool8_t` [binaryFile](#)
Whether to save the PPM as a binary file.
- unsigned int [reserved](#) [16]
Reserved for future use.

12.17.1 Detailed Description

Options for saving PPM images.

Used in saving PPM images with a call to [spinImageSavePpm\(\)](#).

12.17.2 Field Documentation

12.17.2.1 `binaryFile`

```
bool8_t binaryFile
```

Whether to save the PPM as a binary file.

12.17.2.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/spinc/[SpinnakerDefsC.h](#)

12.18 _spinTIFFOption Struct Reference

Options for saving TIFF images.

Data Fields

- spinTIFFCompressionMethod [compression](#)
Compression method to use for encoding TIFF images.
- unsigned int [reserved](#) [16]
Reserved for future use.

12.18.1 Detailed Description

Options for saving TIFF images.

Used in saving PPM images with a call to [spinImageSaveTiff\(\)](#).

12.18.2 Field Documentation

12.18.2.1 compression

```
spinTIFFCompressionMethod compression
```

Compression method to use for encoding TIFF images.

12.18.2.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/spinc/[SpinnakerDefsC.h](#)

Chapter 13

File Documentation

13.1 doc/spindocs/C/GettingStarted.dox File Reference

13.2 doc/spindocs/C/ProgrammerGuide.dox File Reference

13.3 doc/spindocs/shared/Benefits.dox File Reference

13.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference

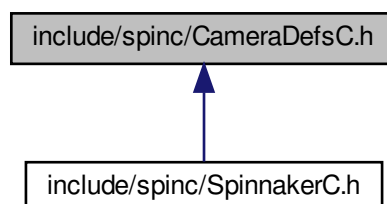
13.5 doc/spindocs/shared/GenICamGenTL.dox File Reference

13.6 doc/spindocs/shared/Licensing.dox File Reference

13.7 doc/spindocs/shared/Maintenance.dox File Reference

13.8 include/spinc/CameraDefsC.h File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

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

The enum definitions for camera nodes.

- enum `_spinExposureModeEnums` {
`ExposureMode_Timed`,
`ExposureMode_TriggerWidth`,
`NUM_EXPOSUREMODE` }
- enum `_spinAcquisitionModeEnums` {
`AcquisitionMode_Continuous`,
`AcquisitionMode_SingleFrame`,
`AcquisitionMode_MultiFrame`,
`NUM_ACQUISITIONMODE` }
- enum `_spinTriggerSourceEnums` {
`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 `_spinTriggerActivationEnums` {
`TriggerActivation_LevelLow`,
`TriggerActivation_LevelHigh`,
`TriggerActivation_FallingEdge`,
`TriggerActivation_RisingEdge`,
`TriggerActivation_AnyEdge`,
`NUM_TRIGGERACTIVATION` }
- enum `_spinSensorShutterModeEnums` {
`SensorShutterMode_Global`,
`SensorShutterMode_Rolling`,
`SensorShutterMode_GlobalReset`,
`NUM_SENSORSHUTTERMODE` }
- enum `_spinTriggerModeEnums` {
`TriggerMode_Off`,
`TriggerMode_On`,
`NUM_TRIGGERMODE` }
- enum `_spinTriggerOverlapEnums` {
`TriggerOverlap_Off`,
`TriggerOverlap_ReadOut`,
`TriggerOverlap_PreviousFrame`,
`NUM_TRIGGEROVERLAP` }
- enum `_spinTriggerSelectorEnums` {
`TriggerSelector_AcquisitionStart`,
`TriggerSelector_FrameStart`,

```

    TriggerSelector_FrameBurstStart,
    NUM_TRIGGERSELECTOR }

• enum _spinExposureAutoEnums {
    ExposureAuto_Off,
    ExposureAuto_Once,
    ExposureAuto_Continuous,
    NUM_EXPOSUREAUTO }

• enum _spinEventSelectorEnums {
    EventSelector_Error,
    EventSelector_ExposureEnd,
    EventSelector_SerialPortReceive,
    NUM_EVENTSELECTOR }

• enum _spinEventNotificationEnums {
    EventNotification_On,
    EventNotification_Off,
    NUM_EVENTNOTIFICATION }

• enum _spinLogicBlockSelectorEnums {
    LogicBlockSelector_LogicBlock0,
    LogicBlockSelector_LogicBlock1,
    NUM_LOGICBLOCKSELECTOR }

• enum _spinLogicBlockLUTInputActivationEnums {
    LogicBlockLUTInputActivation_LevelLow,
    LogicBlockLUTInputActivation_LevelHigh,
    LogicBlockLUTInputActivation_FallingEdge,
    LogicBlockLUTInputActivation_RisingEdge,
    LogicBlockLUTInputActivation_AnyEdge,
    NUM_LOGICBLOCKLUTINPUTACTIVATION }

• enum _spinLogicBlockLUTInputSelectorEnums {
    LogicBlockLUTInputSelector_Input0,
    LogicBlockLUTInputSelector_Input1,
    LogicBlockLUTInputSelector_Input2,
    LogicBlockLUTInputSelector_Input3,
    NUM_LOGICBLOCKLUTINPUTSELECTOR }

• enum _spinLogicBlockLUTInputSourceEnums {
    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 _spinLogicBlockLUTSelectorEnums {
    LogicBlockLUTSelector_Value,
    LogicBlockLUTSelector_Enable,
    NUM_LOGICBLOCKLUTSELECTOR }

```

- `enum _spinColorTransformationSelectorEnums {`
`ColorTransformationSelector_RGBtoRGB,`
`ColorTransformationSelector_RGBtoYUV,`
`NUM_COLORTRANSFORMATIONSELECTOR }`
- `enum _spinRgbTransformLightSourceEnums {`
`RgbTransformLightSource_General,`
`RgbTransformLightSource_Tungsten2800K,`
`RgbTransformLightSource_WarmFluorescent3000K,`
`RgbTransformLightSource_CoolFluorescent4000K,`
`RgbTransformLightSource_Daylight5000K,`
`RgbTransformLightSource_Cloudy6500K,`
`RgbTransformLightSource_Shade8000K,`
`RgbTransformLightSource_Custom,`
`NUM_RGBTRANSFORMLIGHTSOURCE }`
- `enum _spinColorTransformationValueSelectorEnums {`
`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 _spinDeviceRegistersEndiannessEnums {`
`DeviceRegistersEndianness_Little,`
`DeviceRegistersEndianness_Big,`
`NUM_DEVICEREGISTERSENDIANNESSE }`
- `enum _spinDeviceScanTypeEnums {`
`DeviceScanType_Areascan,`
`NUM_DEVICESCANTYPE }`
- `enum _spinDeviceCharacterSetEnums {`
`DeviceCharacterSet_UTF8,`
`DeviceCharacterSet_ASCII,`
`NUM_DEVICECHARACTERSET }`
- `enum _spinDeviceTLTypeEnums {`
`DeviceTLType_GigEVision,`
`DeviceTLType_CameraLink,`
`DeviceTLType_CameraLinkHS,`
`DeviceTLType_CoaXPRESS,`
`DeviceTLType_USB3Vision,`
`DeviceTLType_Custom,`
`NUM_DEVICETLTYPE }`
- `enum _spinDevicePowerSupplySelectorEnums {`
`DevicePowerSupplySelector_External,`
`NUM_DEVICEPOWERSUPPLYSELECTOR }`
- `enum _spinDeviceTemperatureSelectorEnums {`
`DeviceTemperatureSelector_Sensor,`
`NUM_DEVICETEMPERATURESELECTOR }`
- `enum _spinDeviceIndicatorModeEnums {`
`DeviceIndicatorMode_Inactive,`
`DeviceIndicatorMode_Active,`
`DeviceIndicatorMode_ErrorStatus,`
`NUM_DEVICEINDICATORMODE }`

- enum `_spinAutoExposureControlPriorityEnums` {
`AutoExposureControlPriority_Gain,`
`AutoExposureControlPriority_ExposureTime,`
`NUM_AUTOEXPOSURECONTROLPRIORITY` }
- enum `_spinAutoExposureMeteringModeEnums` {
`AutoExposureMeteringMode_Average,`
`AutoExposureMeteringMode_Spot,`
`AutoExposureMeteringMode_Partial,`
`AutoExposureMeteringMode_CenterWeighted,`
`AutoExposureMeteringMode_HistogramPeak,`
`NUM_AUTOEXPOSUREMETERINGMODE` }
- enum `_spinBalanceWhiteAutoProfileEnums` {
`BalanceWhiteAutoProfile_Indoor,`
`BalanceWhiteAutoProfile_Outdoor,`
`NUM_BALANCEWHITEAUTOPROFILE` }
- enum `_spinAutoAlgorithmSelectorEnums` {
`AutoAlgorithmSelector_Awb,`
`AutoAlgorithmSelector_Ae,`
`NUM_AUTOALGORITHMSELECTOR` }
- enum `_spinAutoExposureTargetGreyValueAutoEnums` {
`AutoExposureTargetGreyValueAuto_Off,`
`AutoExposureTargetGreyValueAuto_Continuous,`
`NUM_AUTOEXPOSURETARGETGREYVALUEAUTO` }
- enum `_spinAutoExposureLightingModeEnums` {
`AutoExposureLightingMode_AutoDetect,`
`AutoExposureLightingMode_Backlight,`
`AutoExposureLightingMode_Frontlight,`
`AutoExposureLightingMode_Normal,`
`NUM_AUTOEXPOSURELIGHTINGMODE` }
- enum `_spinGevIEEE1588StatusEnums` {
`GevIEEE1588Status_Initializing,`
`GevIEEE1588Status_Faulty,`
`GevIEEE1588Status_Disabled,`
`GevIEEE1588Status_Listening,`
`GevIEEE1588Status_PreMaster,`
`GevIEEE1588Status_Master,`
`GevIEEE1588Status_Passive,`
`GevIEEE1588Status_Uncalibrated,`
`GevIEEE1588Status_Slave,`
`NUM_GEVIEEE1588STATUS` }
- enum `_spinGevIEEE1588ModeEnums` {
`GevIEEE1588Mode_Auto,`
`GevIEEE1588Mode_SlaveOnly,`
`NUM_GEVIEEE1588MODE` }
- enum `_spinGevIEEE1588ClockAccuracyEnums` {
`GevIEEE1588ClockAccuracy_Unknown,`
`NUM_GEVIEEE1588CLOCKACCURACY` }
- enum `_spinGevCCPEnums` {
`GevCCP_OpenAccess,`
`GevCCP_ExclusiveAccess,`
`GevCCP_ControlAccess,`
`NUM_GEVCCP` }
- enum `_spinGevSupportedOptionSelectorEnums` {
`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 _spinBlackLevelSelectorEnums {
    BlackLevelSelector_All,
    BlackLevelSelector_Analog,
    BlackLevelSelector_Digital,
    NUM_BLACKLEVELSELECTOR }
• enum _spinBalanceWhiteAutoEnums {
    BalanceWhiteAuto_Off,
    BalanceWhiteAuto_Once,
    BalanceWhiteAuto_Continuous,
    NUM_BALANCEWHITEAUTO }
• enum _spinGainAutoEnums {
    GainAuto_Off,
    GainAuto_Once,
    GainAuto_Continuous,
    NUM_GAINAUTO }
• enum _spinBalanceRatioSelectorEnums {
    BalanceRatioSelector_Red,
    BalanceRatioSelector_Blue,
    NUM_BALANCERATIOSELECTOR }
• enum _spinGainSelectorEnums {
    GainSelector_All,
    NUM_GAINSELECTOR }
• enum _spinDefectCorrectionModeEnums {
    DefectCorrectionMode_Average,
    DefectCorrectionMode_Highlight,
    DefectCorrectionMode_Zero,
    NUM_DEFECTCORRECTIONMODE }
• enum _spinUserSetSelectorEnums {
    UserSetSelector_Default,
    UserSetSelector_UserSet0,
    UserSetSelector_UserSet1,
    NUM_USERSETSELECTOR }
• enum _spinUserSetDefaultEnums {
    UserSetDefault_Default,
    UserSetDefault_UserSet0,
    UserSetDefault_UserSet1,
    NUM_USERSETDEFAULT }
• enum _spinSerialPortBaudRateEnums {
    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 _spinSerialPortParityEnums {
    SerialPortParity_None,
    SerialPortParity_Odd,
    SerialPortParity_Even,
    SerialPortParity_Mark,
    SerialPortParity_Space,
    NUM_SERIALPORTPARITY }

• enum _spinSerialPortSelectorEnums {
    SerialPortSelector_SerialPort0,
    NUM_SERIALPORTSELECTOR }

• enum _spinSerialPortStopBitsEnums {
    SerialPortStopBits_Bits1,
    SerialPortStopBits_Bits1AndAHalf,
    SerialPortStopBits_Bits2,
    NUM_SERIALPORTSTOPBITS }

• enum _spinSerialPortSourceEnums {
    SerialPortSource_Line0,
    SerialPortSource_Line1,
    SerialPortSource_Line2,
    SerialPortSource_Line3,
    SerialPortSource_Off,
    NUM_SERIALPORTSOURCE }

• enum _spinSequencerModeEnums {
    SequencerMode_Off,
    SequencerMode_On,
    NUM_SEQUENCERMODE }

• enum _spinSequencerConfigurationValidEnums {
    SequencerConfigurationValid_No,
    SequencerConfigurationValid_Yes,
    NUM_SEQUENCERCONFIGURATIONVALID }

• enum _spinSequencerSetValidEnums {
    SequencerSetValid_No,
    SequencerSetValid_Yes,
    NUM_SEQUENCERSETVALID }

• enum _spinSequencerTriggerActivationEnums {
    SequencerTriggerActivation_RisingEdge,
    SequencerTriggerActivation_FallingEdge,
    SequencerTriggerActivation_AnyEdge,
    SequencerTriggerActivation_LevelHigh,
    SequencerTriggerActivation_LevelLow,
    NUM_SEQUENCERTRIGGERACTIVATION }

• enum _spinSequencerConfigurationModeEnums {
    SequencerConfigurationMode_Off,
    SequencerConfigurationMode_On,
    NUM_SEQUENCERCONFIGURATIONMODE }

```

- enum `_spinSequencerTriggerSourceEnums` {
`SequencerTriggerSource_Off`,
`SequencerTriggerSource_FrameStart`,
`NUM_SEQUENCERTRIGGERSOURCE` }
- enum `_spinTransferQueueModeEnums` {
`TransferQueueMode_FirstInFirstOut`,
`NUM_TRANSFERQUEUEMODE` }
- enum `_spinTransferOperationModeEnums` {
`TransferOperationMode_Continuous`,
`TransferOperationMode_MultiBlock`,
`NUM_TRANSFEROPERATIONMODE` }
- enum `_spinTransferControlModeEnums` {
`TransferControlMode_Basic`,
`TransferControlMode_Automatic`,
`TransferControlMode_UserControlled`,
`NUM_TRANSFERCONTROLMODE` }
- enum `_spinChunkGainSelectorEnums` {
`ChunkGainSelector_All`,
`ChunkGainSelector_Red`,
`ChunkGainSelector_Green`,
`ChunkGainSelector_Blue`,
`NUM_CHUNKGAINSELECTOR` }
- enum `_spinChunkSelectorEnums` {
`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 `_spinChunkBlackLevelSelectorEnums` {
`ChunkBlackLevelSelector_All`,
`NUM_CHUNKBLACKLEVELSELECTOR` }
- enum `_spinChunkPixelFormatEnums` {
`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 `_spinFileOperationStatusEnums` {
`FileOperationStatus_Success`,
`FileOperationStatus_Failure`,

```

    FileOperationStatus_Overflow,
    NUM_FILEOPERATIONSTATUS }
• enum _spinFileOpenModeEnums {
    FileOpenMode_Read,
    FileOpenMode_Write,
    FileOpenMode_ReadWrite,
    NUM_FILEOPENMODE }
• enum _spinFileOperationSelectorEnums {
    FileOperationSelector_Open,
    FileOperationSelector_Close,
    FileOperationSelector_Read,
    FileOperationSelector_Write,
    FileOperationSelector_Delete,
    NUM_FILEOPERATIONSELECTOR }
• enum _spinFileSelectorEnums {
    FileSelector_UserSetDefault,
    FileSelector_UserSet0,
    FileSelector_UserSet1,
    FileSelector_UserFile1,
    FileSelector_SerialPort0,
    NUM_FILESELECTOR }
• enum _spinBinningSelectorEnums {
    BinningSelector_All,
    BinningSelector_Sensor,
    BinningSelector_ISP,
    NUM_BINNINGSELECTOR }
• enum _spinTestPatternGeneratorSelectorEnums {
    TestPatternGeneratorSelector_Sensor,
    TestPatternGeneratorSelector_PipelineStart,
    NUM_TESTPATTERNGENERATORSELECTOR }
• enum _spinCompressionSaturationPriorityEnums {
    CompressionSaturationPriority_DropFrame,
    CompressionSaturationPriority_ReduceFrameRate,
    NUM_COMPRESSIONSATURATIONPRIORITY }
• enum _spinTestPatternEnums {
    TestPattern_Off,
    TestPattern_Increment,
    TestPattern_SensorTestPattern,
    NUM_TESTPATTERN }
• enum _spinPixelColorFilterEnums {
    PixelColorFilter_None,
    PixelColorFilter_BayerRG,
    PixelColorFilter_BayerGB,
    PixelColorFilter_BayerGR,
    PixelColorFilter_BayerBG,
    NUM_PIXELCOLORFILTER }
• enum _spinAdcBitDepthEnums {
    AdcBitDepth_Bit8,
    AdcBitDepth_Bit10,
    AdcBitDepth_Bit12,
    AdcBitDepth_Bit14,
    NUM_ADCBITDEPTH }
• enum _spinDecimationHorizontalModeEnums {
    DecimationHorizontalMode_Discard,
    NUM_DECIMATIONHORIZONTALMODE }
• enum _spinBinningVerticalModeEnums {
    BinningVerticalMode_Sum,

```

```

    BinningVerticalMode_Average,
    NUM_BINNINGVERTICALMODE }
• enum _spinPixelSizeEnums {
    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 _spinDecimationSelectorEnums {
    DecimationSelector_All,
    DecimationSelector_Sensor,
    NUM_DECIMATIONSELECTOR }
• enum _spinImageCompressionModeEnums {
    ImageCompressionMode_Off,
    ImageCompressionMode_Lossless,
    NUM_IMAGECOMPRESSIONMODE }
• enum _spinBinningHorizontalModeEnums {
    BinningHorizontalMode_Sum,
    BinningHorizontalMode_Average,
    NUM_BINNINGHORIZONTALMODE }
• enum _spinPixelFormatEnums {
    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_Mono16s,
PixelFormat_Mono32f,
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,
PixelFormat_RGB10p,
PixelFormat_RGB10p32,
PixelFormat_RGB12,
PixelFormat_RGB12_Planar,
PixelFormat_RGB12p,
PixelFormat_RGB14,
PixelFormat_RGB16,
PixelFormat_RGB16s,
PixelFormat_RGB32f,
PixelFormat_RGB16_Planar,
PixelFormat_RGB565p,
PixelFormat_BGRa10,
PixelFormat_BGRa10p,
PixelFormat_BGRa12,
PixelFormat_BGRa12p,
PixelFormat_BGRa14,

PixelFormat_BGRa16,
PixelFormat_RGBa32f,
PixelFormat_BGR10,
PixelFormat_BGR10p,
PixelFormat_BGR12,
PixelFormat_BGR12p,
PixelFormat_BGR14,
PixelFormat_BGR16,
PixelFormat_BGR565p,
PixelFormat_R8,
PixelFormat_R10,
PixelFormat_R12,
PixelFormat_R16,
PixelFormat_G8,
PixelFormat_G10,
PixelFormat_G12,
PixelFormat_G16,
PixelFormat_B8,
PixelFormat_B10,
PixelFormat_B12,
PixelFormat_B16,
PixelFormat_Coord3D_ABC8,
PixelFormat_Coord3D_ABC8_Planar,
PixelFormat_Coord3D_ABC10p,
PixelFormat_Coord3D_ABC10p_Planar,
PixelFormat_Coord3D_ABC12p,
PixelFormat_Coord3D_ABC12p_Planar,
PixelFormat_Coord3D_ABC16,
PixelFormat_Coord3D_ABC16_Planar,
PixelFormat_Coord3D_ABC32f,
PixelFormat_Coord3D_ABC32f_Planar,
PixelFormat_Coord3D_AC8,
PixelFormat_Coord3D_AC8_Planar,
PixelFormat_Coord3D_AC10p,
PixelFormat_Coord3D_AC10p_Planar,
PixelFormat_Coord3D_AC12p,
PixelFormat_Coord3D_AC12p_Planar,
PixelFormat_Coord3D_AC16,
PixelFormat_Coord3D_AC16_Planar,
PixelFormat_Coord3D_AC32f,
PixelFormat_Coord3D_AC32f_Planar,
PixelFormat_Coord3D_A8,
PixelFormat_Coord3D_A10p,
PixelFormat_Coord3D_A12p,
PixelFormat_Coord3D_A16,
PixelFormat_Coord3D_A32f,
PixelFormat_Coord3D_B8,
PixelFormat_Coord3D_B10p,
PixelFormat_Coord3D_B12p,
PixelFormat_Coord3D_B16,
PixelFormat_Coord3D_B32f,
PixelFormat_Coord3D_C8,
PixelFormat_Coord3D_C10p,
PixelFormat_Coord3D_C12p,
PixelFormat_Coord3D_C16,
PixelFormat_Coord3D_C32f,
PixelFormat_Confidence1,
PixelFormat_Confidence1p,

PixelFormat_Confidence8,
PixelFormat_Confidence16,
PixelFormat_Confidence32f,
PixelFormat_BiColorBGRG8,
PixelFormat_BiColorBGRG10,
PixelFormat_BiColorBGRG10p,
PixelFormat_BiColorBGRG12,
PixelFormat_BiColorBGRG12p,
PixelFormat_BiColorRGBG8,
PixelFormat_BiColorRGBG10,
PixelFormat_BiColorRGBG10p,
PixelFormat_BiColorRGBG12,
PixelFormat_BiColorRGBG12p,
PixelFormat_SCF1WBWG8,
PixelFormat_SCF1WBWG10,
PixelFormat_SCF1WBWG10p,
PixelFormat_SCF1WBWG12,
PixelFormat_SCF1WBWG12p,
PixelFormat_SCF1WBWG14,
PixelFormat_SCF1WBWG16,
PixelFormat_SCF1WGWB8,
PixelFormat_SCF1WGWB10,
PixelFormat_SCF1WGWB10p,
PixelFormat_SCF1WGWB12,
PixelFormat_SCF1WGWB12p,
PixelFormat_SCF1WGWB14,
PixelFormat_SCF1WGWB16,
PixelFormat_SCF1WGWR8,
PixelFormat_SCF1WGWR10,
PixelFormat_SCF1WGWR10p,
PixelFormat_SCF1WGWR12,
PixelFormat_SCF1WGWR12p,
PixelFormat_SCF1WGWR14,
PixelFormat_SCF1WGWR16,
PixelFormat_SCF1WRWG8,
PixelFormat_SCF1WRWG10,
PixelFormat_SCF1WRWG10p,
PixelFormat_SCF1WRWG12,
PixelFormat_SCF1WRWG12p,
PixelFormat_SCF1WRWG14,
PixelFormat_SCF1WRWG16,
PixelFormat_YCbCr8_CbYCr,
PixelFormat_YCbCr10_CbYCr,
PixelFormat_YCbCr10p_CbYCr,
PixelFormat_YCbCr12_CbYCr,
PixelFormat_YCbCr12p_CbYCr,
PixelFormat_YCbCr411_8_CbYYCrYY,
PixelFormat_YCbCr422_8_CbYCrY,
PixelFormat_YCbCr422_10,
PixelFormat_YCbCr422_10_CbYCrY,
PixelFormat_YCbCr422_10p,
PixelFormat_YCbCr422_10p_CbYCrY,
PixelFormat_YCbCr422_12,
PixelFormat_YCbCr422_12_CbYCrY,
PixelFormat_YCbCr422_12p,
PixelFormat_YCbCr422_12p_CbYCrY,
PixelFormat_YCbCr601_8_CbYCr,
PixelFormat_YCbCr601_10_CbYCr,

```

PixelFormat_YCbCr601_10p_CbYCr,
PixelFormat_YCbCr601_12_CbYCr,
PixelFormat_YCbCr601_12p_CbYCr,
PixelFormat_YCbCr601_411_8_CbYYCrYY,
PixelFormat_YCbCr601_422_8,
PixelFormat_YCbCr601_422_8_CbYCrY,
PixelFormat_YCbCr601_422_10,
PixelFormat_YCbCr601_422_10_CbYCrY,
PixelFormat_YCbCr601_422_10p,
PixelFormat_YCbCr601_422_10p_CbYCrY,
PixelFormat_YCbCr601_422_12,
PixelFormat_YCbCr601_422_12_CbYCrY,
PixelFormat_YCbCr601_422_12p,
PixelFormat_YCbCr601_422_12p_CbYCrY,
PixelFormat_YCbCr709_8_CbYCr,
PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,
PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,
PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,
PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,
PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
PixelFormat_GR12,
PixelFormat_GB12,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

```

- enum _spinDecimationVerticalModeEnums {
DecimationVerticalMode_Discard,


```

    NUM_DECIMATIONVERTICALMODE }
• enum _spinLineModeEnums {
    LineMode_Input,
    LineMode_Output,
    NUM_LINEMODE }
• enum _spinLineSourceEnums {
    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 _spinLineInputFilterSelectorEnums {
    LineInputFilterSelector_Deglintch,
    LineInputFilterSelector_Debounce,
    NUM_LINEINPUTFILTERSELECTOR }
• enum _spinUserOutputSelectorEnums {
    UserOutputSelector_UserOutput0,
    UserOutputSelector_UserOutput1,
    UserOutputSelector_UserOutput2,
    UserOutputSelector_UserOutput3,
    NUM_USEROUTPUTSELECTOR }
• enum _spinLineFormatEnums {
    LineFormat_NoConnect,
    LineFormat_TriState,
    LineFormat_TTL,
    LineFormat_LVDS,
    LineFormat_RS422,
    LineFormat_OptoCoupled,
    LineFormat_OpenDrain,
    NUM_LINEFORMAT }
• enum _spinLineSelectorEnums {
    LineSelector_Line0,
    LineSelector_Line1,
    LineSelector_Line2,
    LineSelector_Line3,
    NUM_LINESELECTOR }
• enum _spinExposureActiveModeEnums {
    ExposureActiveMode_Line1,
    ExposureActiveMode_AnyPixels,
    ExposureActiveMode_AllPixels,
    NUM_EXPOSUREACTIVEMODE }
• enum _spinCounterTriggerActivationEnums {
    CounterTriggerActivation_LevelLow,

```

```

CounterTriggerActivation_LevelHigh,
CounterTriggerActivation_FallingEdge,
CounterTriggerActivation_RisingEdge,
CounterTriggerActivation_AnyEdge,
NUM_COUNTERTRIGGERACTIVATION }

• enum _spinCounterSelectorEnums {
CounterSelector_Counter0,
CounterSelector_Counter1,
NUM_COUNTERSELECTOR }

• enum _spinCounterStatusEnums {
CounterStatus_CounterIdle,
CounterStatus_CounterTriggerWait,
CounterStatus_CounterActive,
CounterStatus_CounterCompleted,
CounterStatus_CounterOverflow,
NUM_COUNTERSTATUS }

• enum _spinCounterTriggerSourceEnums {
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 _spinCounterResetSourceEnums {
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 _spinCounterEventSourceEnums {
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 _spinCounterEventActivationEnums {
CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }

• enum _spinCounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }

• enum _spinDeviceTypeEnums {
DeviceType_Transmitter,
DeviceType_Receiver,
DeviceType_Transceiver,
DeviceType_Peripheral,
NUM_DEVICEYPE }

• enum _spinDeviceConnectionStatusEnums {
DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,
NUM_DEVICECONNECTIONSTATUS }

• enum _spinDeviceLinkThroughputLimitModeEnums {
DeviceLinkThroughputLimitMode_On,
DeviceLinkThroughputLimitMode_Off,
NUM_DEVICELINKTHROUGHPUTLIMITMODE }

• enum _spinDeviceLinkHeartbeatModeEnums {
DeviceLinkHeartbeatMode_On,
DeviceLinkHeartbeatMode_Off,
NUM_DEVICELINKHEARTBEATMODE }

• enum _spinDeviceStreamChannelTypeEnums {
DeviceStreamChannelType_Transmitter,
DeviceStreamChannelType_Receiver,
NUM_DEVICESTREAMCHANNELTYPE }

• enum _spinDeviceStreamChannelEndiannessEnums {
DeviceStreamChannelEndianness_Big,

```

```

DeviceStreamChannelEndianness_Little,
NUM_DEVICESTREAMCHANNELENDIANNESS }
• enum _spinDeviceClockSelectorEnums {
DeviceClockSelector_Sensor,
DeviceClockSelector_SensorDigitization,
DeviceClockSelector_CameraLink,
NUM_DEVICECLOCKSELECTOR }
• enum _spinDeviceSerialPortSelectorEnums {
DeviceSerialPortSelector_CameraLink,
NUM_DEVICSERIALPORTSELECTOR }
• enum _spinDeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600,
DeviceSerialPortBaudRate_Baud19200,
DeviceSerialPortBaudRate_Baud38400,
DeviceSerialPortBaudRate_Baud57600,
DeviceSerialPortBaudRate_Baud115200,
DeviceSerialPortBaudRate_Baud230400,
DeviceSerialPortBaudRate_Baud460800,
DeviceSerialPortBaudRate_Baud921600,
NUM_DEVICSERIALPORTBAUDRATE }
• enum _spinSensorTapsEnums {
SensorTaps_One,
SensorTaps_Two,
SensorTaps_Three,
SensorTaps_Four,
SensorTaps_Eight,
SensorTaps_Ten,
NUM_SENSORTAPS }
• enum _spinSensorDigitizationTapsEnums {
SensorDigitizationTaps_One,
SensorDigitizationTaps_Two,
SensorDigitizationTaps_Three,
SensorDigitizationTaps_Four,
SensorDigitizationTaps_Eight,
SensorDigitizationTaps_Ten,
NUM_SENSORDIGITIZATIONTAPS }
• enum _spinRegionSelectorEnums {
RegionSelector_Region0,
RegionSelector_Region1,
RegionSelector_Region2,
RegionSelector_All,
NUM_REGIONSELECTOR }
• enum _spinRegionModeEnums {
RegionMode_Off,
RegionMode_On,
NUM_REGIONMODE }
• enum _spinRegionDestinationEnums {
RegionDestination_Stream0,
RegionDestination_Stream1,
RegionDestination_Stream2,
NUM_REGIONDESTINATION }
• enum _spinImageComponentSelectorEnums {
ImageComponentSelector_Intensity,
ImageComponentSelector_Color,
ImageComponentSelector_Infrared,
ImageComponentSelector_Ultraviolet,
ImageComponentSelector_Range,
ImageComponentSelector_Disparity,

```

```
ImageComponentSelector_Confidence,  
ImageComponentSelector_Scatter,  
NUM_IMAGECOMPONENTSELECTOR }
```

- enum _spinPixelFormatInfoSelectorEnums {
PixelFormatInfoSelector_Mono1p,
PixelFormatInfoSelector_Mono2p,
PixelFormatInfoSelector_Mono4p,
PixelFormatInfoSelector_Mono8,
PixelFormatInfoSelector_Mono8s,
PixelFormatInfoSelector_Mono10,
PixelFormatInfoSelector_Mono10p,
PixelFormatInfoSelector_Mono12,
PixelFormatInfoSelector_Mono12p,
PixelFormatInfoSelector_Mono14,
PixelFormatInfoSelector_Mono16,
PixelFormatInfoSelector_Mono16s,
PixelFormatInfoSelector_Mono32f,
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_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_RGB16s,
PixelFormatInfoSelector_RGB32f,
PixelFormatInfoSelector_RGB16_Planar,
PixelFormatInfoSelector_RGB565p,
PixelFormatInfoSelector_BGRa8,
PixelFormatInfoSelector_BGRa10,
PixelFormatInfoSelector_BGRa10p,
PixelFormatInfoSelector_BGRa12,
PixelFormatInfoSelector_BGRa12p,
PixelFormatInfoSelector_BGRa14,
PixelFormatInfoSelector_BGRa16,
PixelFormatInfoSelector_RGBa32f,
PixelFormatInfoSelector_BGR8,
PixelFormatInfoSelector_BGR10,
PixelFormatInfoSelector_BGR10p,
PixelFormatInfoSelector_BGR12,
PixelFormatInfoSelector_BGR12p,
PixelFormatInfoSelector_BGR14,
PixelFormatInfoSelector_BGR16,
PixelFormatInfoSelector_BGR565p,
PixelFormatInfoSelector_R8,
PixelFormatInfoSelector_R10,
PixelFormatInfoSelector_R12,
PixelFormatInfoSelector_R16,
PixelFormatInfoSelector_G8,
PixelFormatInfoSelector_G10,
PixelFormatInfoSelector_G12,
PixelFormatInfoSelector_G16,
PixelFormatInfoSelector_B8,
PixelFormatInfoSelector_B10,
PixelFormatInfoSelector_B12,
PixelFormatInfoSelector_B16,
PixelFormatInfoSelector_Coord3D_ABC8,
PixelFormatInfoSelector_Coord3D_ABC8_Planar,
PixelFormatInfoSelector_Coord3D_ABC10p,
PixelFormatInfoSelector_Coord3D_ABC10p_Planar,
PixelFormatInfoSelector_Coord3D_ABC12p,
PixelFormatInfoSelector_Coord3D_ABC12p_Planar,
PixelFormatInfoSelector_Coord3D_ABC16,
PixelFormatInfoSelector_Coord3D_ABC16_Planar,
PixelFormatInfoSelector_Coord3D_ABC32f,
PixelFormatInfoSelector_Coord3D_ABC32f_Planar,
PixelFormatInfoSelector_Coord3D_AC8,
PixelFormatInfoSelector_Coord3D_AC8_Planar,
PixelFormatInfoSelector_Coord3D_AC10p,
PixelFormatInfoSelector_Coord3D_AC10p_Planar,
PixelFormatInfoSelector_Coord3D_AC12p,
PixelFormatInfoSelector_Coord3D_AC12p_Planar,
PixelFormatInfoSelector_Coord3D_AC16,
PixelFormatInfoSelector_Coord3D_AC16_Planar,
PixelFormatInfoSelector_Coord3D_AC32f,
PixelFormatInfoSelector_Coord3D_AC32f_Planar,
PixelFormatInfoSelector_Coord3D_A8,
PixelFormatInfoSelector_Coord3D_A10p,
PixelFormatInfoSelector_Coord3D_A12p,
PixelFormatInfoSelector_Coord3D_A16,
PixelFormatInfoSelector_Coord3D_A32f,

PixelFormatInfoSelector_Coord3D_B8,
PixelFormatInfoSelector_Coord3D_B10p,
PixelFormatInfoSelector_Coord3D_B12p,
PixelFormatInfoSelector_Coord3D_B16,
PixelFormatInfoSelector_Coord3D_B32f,
PixelFormatInfoSelector_Coord3D_C8,
PixelFormatInfoSelector_Coord3D_C10p,
PixelFormatInfoSelector_Coord3D_C12p,
PixelFormatInfoSelector_Coord3D_C16,
PixelFormatInfoSelector_Coord3D_C32f,
PixelFormatInfoSelector_Confidence1,
PixelFormatInfoSelector_Confidence1p,
PixelFormatInfoSelector_Confidence8,
PixelFormatInfoSelector_Confidence16,
PixelFormatInfoSelector_Confidence32f,
PixelFormatInfoSelector_BiColorBGRG8,
PixelFormatInfoSelector_BiColorBGRG10,
PixelFormatInfoSelector_BiColorBGRG10p,
PixelFormatInfoSelector_BiColorBGRG12,
PixelFormatInfoSelector_BiColorBGRG12p,
PixelFormatInfoSelector_BiColorRGBG8,
PixelFormatInfoSelector_BiColorRGBG10,
PixelFormatInfoSelector_BiColorRGBG10p,
PixelFormatInfoSelector_BiColorRGBG12,
PixelFormatInfoSelector_BiColorRGBG12p,
PixelFormatInfoSelector_SCF1WBWG8,
PixelFormatInfoSelector_SCF1WBWG10,
PixelFormatInfoSelector_SCF1WBWG10p,
PixelFormatInfoSelector_SCF1WBWG12,
PixelFormatInfoSelector_SCF1WBWG12p,
PixelFormatInfoSelector_SCF1WBWG14,
PixelFormatInfoSelector_SCF1WBWG16,
PixelFormatInfoSelector_SCF1WGWB8,
PixelFormatInfoSelector_SCF1WGWB10,
PixelFormatInfoSelector_SCF1WGWB10p,
PixelFormatInfoSelector_SCF1WGWB12,
PixelFormatInfoSelector_SCF1WGWB12p,
PixelFormatInfoSelector_SCF1WGWB14,
PixelFormatInfoSelector_SCF1WGWB16,
PixelFormatInfoSelector_SCF1WGWR8,
PixelFormatInfoSelector_SCF1WGWR10,
PixelFormatInfoSelector_SCF1WGWR10p,
PixelFormatInfoSelector_SCF1WGWR12,
PixelFormatInfoSelector_SCF1WGWR12p,
PixelFormatInfoSelector_SCF1WGWR14,
PixelFormatInfoSelector_SCF1WGWR16,
PixelFormatInfoSelector_SCF1WRWG8,
PixelFormatInfoSelector_SCF1WRWG10,
PixelFormatInfoSelector_SCF1WRWG10p,
PixelFormatInfoSelector_SCF1WRWG12,
PixelFormatInfoSelector_SCF1WRWG12p,
PixelFormatInfoSelector_SCF1WRWG14,
PixelFormatInfoSelector_SCF1WRWG16,
PixelFormatInfoSelector_YCbCr8,
PixelFormatInfoSelector_YCbCr8_CbYCr,
PixelFormatInfoSelector_YCbCr10_CbYCr,
PixelFormatInfoSelector_YCbCr10p_CbYCr,
PixelFormatInfoSelector_YCbCr12_CbYCr,

PixelFormatInfoSelector_YCbCr12p_CbYCr,
PixelFormatInfoSelector_YCbCr411_8,
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr422_8,
PixelFormatInfoSelector_YCbCr422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10,
PixelFormatInfoSelector_YCbCr422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10p,
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12,
PixelFormatInfoSelector_YCbCr422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12p,
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,

- ```

PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

```
- enum `_spinDeinterlacingEnums` {
 

```

Deinterlacing_Off,
Deinterlacing_LineDuplication,
Deinterlacing_Weave,
NUM_DEINTERLACING }

```
  - enum `_spinImageCompressionRateOptionEnums` {
 

```

ImageCompressionRateOption_FixBitrate,
ImageCompressionRateOption_FixQuality,
NUM_IMAGECOMPRESSIONRATEOPTION }

```
  - enum `_spinImageCompressionJPEGFormatOptionEnums` {
 

```

ImageCompressionJPEGFormatOption_Lossless,
ImageCompressionJPEGFormatOption_BaselineStandard,
ImageCompressionJPEGFormatOption_BaselineOptimized,
ImageCompressionJPEGFormatOption_Progressive,
NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }

```
  - enum `_spinAcquisitionStatusSelectorEnums` {
 

```

AcquisitionStatusSelector_AcquisitionTriggerWait,
AcquisitionStatusSelector_AcquisitionActive,
AcquisitionStatusSelector_AcquisitionTransfer,
AcquisitionStatusSelector_FrameTriggerWait,
AcquisitionStatusSelector_FrameActive,
AcquisitionStatusSelector_ExposureActive,
NUM_ACQUISITIONSTATUSSELECTOR }

```
  - enum `_spinExposureTimeModeEnums` {
 

```

ExposureTimeMode_Common,
ExposureTimeMode_Individual,
NUM_EXPOSURETIMEMODE }

```
  - enum `_spinExposureTimeSelectorEnums` {
 

```

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 `_spinGainAutoBalanceEnums` {
 

```

GainAutoBalance_Off,
GainAutoBalance_Once,
GainAutoBalance_Continuous,
NUM_GAINAUTOBALANCE }

```
  - enum `_spinBlackLevelAutoEnums` {
 

```

BlackLevelAuto_Off,
BlackLevelAuto_Once,
BlackLevelAuto_Continuous,
NUM_BLACKLEVELAUTO }

```
  - enum `_spinBlackLevelAutoBalanceEnums` {
 

```

BlackLevelAutoBalance_Off,
BlackLevelAutoBalance_Once,

```

```

 BlackLevelAutoBalance_Continuous,
 NUM_BLACKLEVELAUTOBALANCE }
• enum _spinWhiteClipSelectorEnums {
 WhiteClipSelector_All,
 WhiteClipSelector_Red,
 WhiteClipSelector_Green,
 WhiteClipSelector_Blue,
 WhiteClipSelector_Y,
 WhiteClipSelector_U,
 WhiteClipSelector_V,
 WhiteClipSelector_Tap1,
 WhiteClipSelector_Tap2,
 NUM_WHITECLIPSELECTOR }
• enum _spinTimerSelectorEnums {
 TimerSelector_Timer0,
 TimerSelector_Timer1,
 TimerSelector_Timer2,
 NUM_TIMERSELECTOR }
• enum _spinTimerStatusEnums {
 TimerStatus_TimerIdle,
 TimerStatus_TimerTriggerWait,
 TimerStatus_TimerActive,
 TimerStatus_TimerCompleted,
 NUM_TIMERSTATUS }
• enum _spinTimerTriggerSourceEnums {
 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 _spinTimerTriggerActivationEnums {
TimerTriggerActivation_RisingEdge,
TimerTriggerActivation_FallingEdge,
TimerTriggerActivation_AnyEdge,
TimerTriggerActivation_LevelHigh,
TimerTriggerActivation_LevelLow,
NUM_TIMERTRIGGERACTIVATION }
• enum _spinEncoderSelectorEnums {
EncoderSelector_Encoder0,
EncoderSelector_Encoder1,
EncoderSelector_Encoder2,
NUM_ENCODERSELECTOR }
• enum _spinEncoderSourceAEnums {
EncoderSourceA_Off,
EncoderSourceA_Line0,
EncoderSourceA_Line1,
EncoderSourceA_Line2,
NUM_ENCODERSOURCEA }
• enum _spinEncoderSourceBEnums {
EncoderSourceB_Off,
EncoderSourceB_Line0,
EncoderSourceB_Line1,
EncoderSourceB_Line2,
NUM_ENCODERSOURCEB }
• enum _spinEncoderModeEnums {
EncoderMode_FourPhase,
EncoderMode_HighResolution,
NUM_ENCODERMODE }
• enum _spinEncoderOutputModeEnums {
EncoderOutputMode_Off,
EncoderOutputMode_PositionUp,
EncoderOutputMode_PositionDown,
EncoderOutputMode_DirectionUp,
EncoderOutputMode_DirectionDown,
EncoderOutputMode_Motion,
NUM_ENCODEROUTPUTMODE }
• enum _spinEncoderStatusEnums {
EncoderStatus_EncoderUp,
EncoderStatus_EncoderDown,
EncoderStatus_EncoderIdle,
EncoderStatus_EncoderStatic,
NUM_ENCODERSTATUS }
• enum _spinEncoderResetSourceEnums {
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 _spinEncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }

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

• enum _spinActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }

• enum _spinSourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,
NUM_SOURCESELECTOR }

• enum _spinTransferSelectorEnums {
TransferSelector_Stream0,

```

```

TransferSelector_Stream1,
TransferSelector_Stream2,
TransferSelector_All,
NUM_TRANSFERSELECTOR }

• enum _spinTransferTriggerSelectorEnums {
TransferTriggerSelector_TransferStart,
TransferTriggerSelector_TransferStop,
TransferTriggerSelector_TransferAbort,
TransferTriggerSelector_TransferPause,
TransferTriggerSelector_TransferResume,
TransferTriggerSelector_TransferActive,
TransferTriggerSelector_TransferBurstStart,
TransferTriggerSelector_TransferBurstStop,
NUM_TRANSFERTRIGGERSELECTOR }

• enum _spinTransferTriggerModeEnums {
TransferTriggerMode_Off,
TransferTriggerMode_On,
NUM_TRANSFERTRIGGERMODE }

• enum _spinTransferTriggerSourceEnums {
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 _spinTransferTriggerActivationEnums {
TransferTriggerActivation_RisingEdge,
TransferTriggerActivation_FallingEdge,
TransferTriggerActivation_AnyEdge,
TransferTriggerActivation_LevelHigh,
TransferTriggerActivation_LevelLow,
NUM_TRANSFERTRIGGERACTIVATION }

• enum _spinTransferStatusSelectorEnums {
TransferStatusSelector_Streaming,
TransferStatusSelector_Paused,
TransferStatusSelector_Stopping,
TransferStatusSelector_Stopped,
TransferStatusSelector_QueueOverflow,
NUM_TRANSFERSTATUSSELECTOR }

• enum _spinTransferComponentSelectorEnums {
TransferComponentSelector_Red,
TransferComponentSelector_Green,

```

```

TransferComponentSelector_Blue,
TransferComponentSelector_All,
NUM_TRANSFERCOMPONENTSELECTOR }

• enum _spinScan3dDistanceUnitEnums {
Scan3dDistanceUnit_Millimeter,
Scan3dDistanceUnit_Inch,
NUM_SCAN3DDISTANCEUNIT }

• enum _spinScan3dCoordinateSystemEnums {
Scan3dCoordinateSystem_Cartesian,
Scan3dCoordinateSystem_Spherical,
Scan3dCoordinateSystem_Cylindrical,
NUM_SCAN3DCOORDINATESYSTEM }

• enum _spinScan3dOutputModeEnums {
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 _spinScan3dCoordinateSystemReferenceEnums {
Scan3dCoordinateSystemReference_Anchor,
Scan3dCoordinateSystemReference_Transformed,
NUM_SCAN3DCOORDINATESYSTEMREFERENCE }

• enum _spinScan3dCoordinateSelectorEnums {
Scan3dCoordinateSelector_CoordinateA,
Scan3dCoordinateSelector_CoordinateB,
Scan3dCoordinateSelector_CoordinateC,
NUM_SCAN3DCOORDINATESELECTOR }

• enum _spinScan3dCoordinateTransformSelectorEnums {
Scan3dCoordinateTransformSelector_RotationX,
Scan3dCoordinateTransformSelector_RotationY,
Scan3dCoordinateTransformSelector_RotationZ,
Scan3dCoordinateTransformSelector_TranslationX,
Scan3dCoordinateTransformSelector_TranslationY,
Scan3dCoordinateTransformSelector_TranslationZ,
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }

• enum _spinScan3dCoordinateReferenceSelectorEnums {
Scan3dCoordinateReferenceSelector_RotationX,
Scan3dCoordinateReferenceSelector_RotationY,
Scan3dCoordinateReferenceSelector_RotationZ,
Scan3dCoordinateReferenceSelector_TranslationX,
Scan3dCoordinateReferenceSelector_TranslationY,
Scan3dCoordinateReferenceSelector_TranslationZ,
NUM_SCAN3DCOORDINATEREFERENCESELECTOR }

• enum _spinChunkImageComponentEnums {
ChunkImageComponent_Intensity,
ChunkImageComponent_Color,
ChunkImageComponent_Infrared,
ChunkImageComponent_Ultraviolet,
ChunkImageComponent_Range,
ChunkImageComponent_Disparity,
ChunkImageComponent_Confidence,

```

```

 ChunkImageComponent_Scatter,
 NUM_CHUNKIMAGECOMPONENT }
• enum _spinChunkCounterSelectorEnums {
 ChunkCounterSelector_Counter0,
 ChunkCounterSelector_Counter1,
 ChunkCounterSelector_Counter2,
 NUM_CHUNKCOUNTERSELECTOR }
• enum _spinChunkTimerSelectorEnums {
 ChunkTimerSelector_Timer0,
 ChunkTimerSelector_Timer1,
 ChunkTimerSelector_Timer2,
 NUM_CHUNKTIMERSELECTOR }
• enum _spinChunkEncoderSelectorEnums {
 ChunkEncoderSelector_Encoder0,
 ChunkEncoderSelector_Encoder1,
 ChunkEncoderSelector_Encoder2,
 NUM_CHUNKENCODERSELECTOR }
• enum _spinChunkEncoderStatusEnums {
 ChunkEncoderStatus_EncoderUp,
 ChunkEncoderStatus_EncoderDown,
 ChunkEncoderStatus_EncoderIdle,
 ChunkEncoderStatus_EncoderStatic,
 NUM_CHUNKENCODERSTATUS }
• enum _spinChunkExposureTimeSelectorEnums {
 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 _spinChunkSourceIDEnums {
 ChunkSourceID_Source0,
 ChunkSourceID_Source1,
 ChunkSourceID_Source2,
 NUM_CHUNKSOURCEID }
• enum _spinChunkRegionIDEnums {
 ChunkRegionID_Region0,
 ChunkRegionID_Region1,
 ChunkRegionID_Region2,
 NUM_CHUNKREGIONID }
• enum _spinChunkTransferStreamIDEnums {
 ChunkTransferStreamID_Stream0,
 ChunkTransferStreamID_Stream1,
 ChunkTransferStreamID_Stream2,
 ChunkTransferStreamID_Stream3,
 NUM_CHUNKTRANSFERSTREAMID }
• enum _spinChunkScan3dDistanceUnitEnums {
 ChunkScan3dDistanceUnit_Millimeter,
 ChunkScan3dDistanceUnit_Inch,
 NUM_CHUNKSCAN3DDISTANCEUNIT }
• enum _spinChunkScan3dOutputModeEnums {
 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 _spinChunkScan3dCoordinateSystemEnums {
 ChunkScan3dCoordinateSystem_Cartesian,
 ChunkScan3dCoordinateSystem_Spherical,
 ChunkScan3dCoordinateSystem_Cylindrical,
 NUM_CHUNKSCAN3DCOORDINATESYSTEM }

• enum _spinChunkScan3dCoordinateSystemReferenceEnums {
 ChunkScan3dCoordinateSystemReference_Anchor,
 ChunkScan3dCoordinateSystemReference_Transformed,
 NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }

• enum _spinChunkScan3dCoordinateSelectorEnums {
 ChunkScan3dCoordinateSelector_CoordinateA,
 ChunkScan3dCoordinateSelector_CoordinateB,
 ChunkScan3dCoordinateSelector_CoordinateC,
 NUM_CHUNKSCAN3DCOORDINATESELECTOR }

• enum _spinChunkScan3dCoordinateTransformSelectorEnums {
 ChunkScan3dCoordinateTransformSelector_RotationX,
 ChunkScan3dCoordinateTransformSelector_RotationY,
 ChunkScan3dCoordinateTransformSelector_RotationZ,
 ChunkScan3dCoordinateTransformSelector_TranslationX,
 ChunkScan3dCoordinateTransformSelector_TranslationY,
 ChunkScan3dCoordinateTransformSelector_TranslationZ,
 NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }

• enum _spinChunkScan3dCoordinateReferenceSelectorEnums {
 ChunkScan3dCoordinateReferenceSelector_RotationX,
 ChunkScan3dCoordinateReferenceSelector_RotationY,
 ChunkScan3dCoordinateReferenceSelector_RotationZ,
 ChunkScan3dCoordinateReferenceSelector_TranslationX,
 ChunkScan3dCoordinateReferenceSelector_TranslationY,
 ChunkScan3dCoordinateReferenceSelector_TranslationZ,
 NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }

• enum _spinDeviceTapGeometryEnums {
 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 _spinGevPhysicalLinkConfigurationEnums {
 GevPhysicalLinkConfiguration_SingleLink,
 GevPhysicalLinkConfiguration_MultiLink,
 GevPhysicalLinkConfiguration_StaticLAG,
 GevPhysicalLinkConfiguration_DynamicLAG,
 NUM_GEVPHYSICALLINKCONFIGURATION }

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

• enum _spinGevIPConfigurationStatusEnums {
 GevIPConfigurationStatus_None,
 GevIPConfigurationStatus_PersistentIP,
 GevIPConfigurationStatus_DHCP,
 GevIPConfigurationStatus_LLA,
 GevIPConfigurationStatus_ForceIP,
 NUM_GEVIPCONFIGURATIONSTATUS }

• enum _spinGevGVCPExtendedStatusCodesSelectorEnums {
 GevGVCPExtendedStatusCodesSelector_Version1_1,
 GevGVCPExtendedStatusCodesSelector_Version2_0,
 NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }

• enum _spinGevGVSPExtendedIDModeEnums {
 GevGVSPExtendedIDMode_Off,
 GevGVSPExtendedIDMode_On,
 NUM_GEVGVSPEXTENDEDIDMODE }

• enum _spinCICConfigurationEnums {

```

```

CIConfiguration_Base,
CIConfiguration_Medium,
CIConfiguration_Full,
CIConfiguration_DualBase,
CIConfiguration_EightyBit,
NUM_CLCONFIGURATION }

• enum _spinCITimeSlotsCountEnums {
 CITimeSlotsCount_One,
 CITimeSlotsCount_Two,
 CITimeSlotsCount_Three,
 NUM_CLTIMESLOTSCOUNT }

• enum _spinCxpLinkConfigurationStatusEnums {
 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 _spinCxpLinkConfigurationPreferredEnums {
 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 _spinCxpLinkConfigurationEnums {`

```

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 _spinCxpConnectionTestModeEnums {`

```

CxpConnectionTestMode_Off,
CxpConnectionTestMode_Mode1,
NUM_CXPCONNECTIONTESTMODE }

```
 - `enum _spinCxpPoCxpStatusEnums {`

```

CxpPoCxpStatus_Auto,

```

```
CxpPoCxpStatus_Off,
CxpPoCxpStatus_Tripped,
NUM_CXPPOCXPSTATUS }
```

13.8.1 Enumeration Type Documentation

13.8.1.1 _spinAcquisitionModeEnums

```
enum _spinAcquisitionModeEnums
```

< 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	

13.8.1.2 _spinAcquisitionStatusSelectorEnums

```
enum _spinAcquisitionStatusSelectorEnums
```

< 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_ACQUISITIONSTATUSSELECTION	

13.8.1.3 _spinActionUnconditionalModeEnums

enum _spinActionUnconditionalModeEnums

< 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	

13.8.1.4 _spinAdcBitDepthEnums

enum _spinAdcBitDepthEnums

< 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	

13.8.1.5 _spinAutoAlgorithmSelectorEnums

enum _spinAutoAlgorithmSelectorEnums

< 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	

13.8.1.6 `_spinAutoExposureControlPriorityEnums`

```
enum _spinAutoExposureControlPriorityEnums
```

< 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	

13.8.1.7 `_spinAutoExposureLightingModeEnums`

```
enum _spinAutoExposureLightingModeEnums
```

< 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	

13.8.1.8 `_spinAutoExposureMeteringModeEnums`

```
enum _spinAutoExposureMeteringModeEnums
```

< 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	

13.8.1.9 _spinAutoExposureTargetGreyValueAutoEnums

```
enum _spinAutoExposureTargetGreyValueAutoEnums
```

< 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	

13.8.1.10 _spinBalanceRatioSelectorEnums

```
enum _spinBalanceRatioSelectorEnums
```

< 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	

13.8.1.11 _spinBalanceWhiteAutoEnums

```
enum _spinBalanceWhiteAutoEnums
```

< 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	

13.8.1.12 _spinBalanceWhiteAutoProfileEnums

```
enum _spinBalanceWhiteAutoProfileEnums
```

< 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	

13.8.1.13 _spinBinningHorizontalModeEnums

```
enum _spinBinningHorizontalModeEnums
```

<

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	

13.8.1.14 _spinBinningSelectorEnumsenum `_spinBinningSelectorEnums`

< 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	

13.8.1.15 _spinBinningVerticalModeEnumsenum `_spinBinningVerticalModeEnums`

<

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	

13.8.1.16 _spinBlackLevelAutoBalanceEnumsenum `_spinBlackLevelAutoBalanceEnums`

< 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	

13.8.1.17 `_spinBlackLevelAutoEnums`

```
enum _spinBlackLevelAutoEnums
```

< 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	

13.8.1.18 `_spinBlackLevelSelectorEnums`

```
enum _spinBlackLevelSelectorEnums
```

< 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	

13.8.1.19 `_spinChunkBlackLevelSelectorEnums`

```
enum _spinChunkBlackLevelSelectorEnums
```

< Selects which black level to retrieve

Enumerator

ChunkBlackLevelSelector_All	
NUM_CHUNKBLACKLEVELSELECTOR	

13.8.1.20 `_spinChunkCounterSelectorEnums`

```
enum _spinChunkCounterSelectorEnums
```

< 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	

13.8.1.21 _spinChunkEncoderSelectorEnums

```
enum _spinChunkEncoderSelectorEnums
```

< 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	

13.8.1.22 _spinChunkEncoderStatusEnums

```
enum _spinChunkEncoderStatusEnums
```

< 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	

13.8.1.23 _spinChunkExposureTimeSelectorEnums

```
enum _spinChunkExposureTimeSelectorEnums
```

< 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	

13.8.1.24 _spinChunkGainSelectorEnums

```
enum _spinChunkGainSelectorEnums
```

< Selects which gain to retrieve

Enumerator

ChunkGainSelector_All	
ChunkGainSelector_Red	
ChunkGainSelector_Green	
ChunkGainSelector_Blue	
NUM_CHUNKGAINSELECTOR	

13.8.1.25 _spinChunkImageComponentEnums

```
enum _spinChunkImageComponentEnums
```

< 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.

Enumerator

ChunkImageComponent_Confidence	The image data is the confidence map component.
ChunkImageComponent_Scatter	The image data is the scatter component.
NUM_CHUNKIMAGECOMPONENT	

13.8.1.26 _spinChunkPixelFormatEnums

```
enum _spinChunkPixelFormatEnums
```

< 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	

13.8.1.27 _spinChunkRegionIDEnums

```
enum _spinChunkRegionIDEnums
```

< 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	

13.8.1.28 _spinChunkScan3dCoordinateReferenceSelectorEnums

```
enum _spinChunkScan3dCoordinateReferenceSelectorEnums
```

< 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	

13.8.1.29 _spinChunkScan3dCoordinateSelectorEnums

```
enum _spinChunkScan3dCoordinateSelectorEnums
```

< 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	

13.8.1.30 _spinChunkScan3dCoordinateSystemEnums

```
enum _spinChunkScan3dCoordinateSystemEnums
```

< 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	

13.8.1.31 _spinChunkScan3dCoordinateSystemReferenceEnums

```
enum _spinChunkScan3dCoordinateSystemReferenceEnums
```

< 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_CHUNKSCAN3DCOORDINATESYSTEMRE↔ FERENCE	

13.8.1.32 _spinChunkScan3dCoordinateTransformSelectorEnums

```
enum _spinChunkScan3dCoordinateTransformSelectorEnums
```

< 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	

13.8.1.33 _spinChunkScan3dDistanceUnitEnums

```
enum _spinChunkScan3dDistanceUnitEnums
```

< 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	

13.8.1.34 _spinChunkScan3dOutputModeEnums

enum `_spinChunkScan3dOutputModeEnums`

< Returns the Calibrated Mode of the payload image.

Enumerator

<code>ChunkScan3dOutputMode_UncalibratedC</code>	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.
<code>ChunkScan3dOutputMode_CalibratedABC_Grid</code>	3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.
<code>ChunkScan3dOutputMode_CalibratedABC_Point↔ Cloud</code>	3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.
<code>ChunkScan3dOutputMode_CalibratedAC</code>	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.
<code>ChunkScan3dOutputMode_CalibratedAC_Linescan</code>	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.
<code>ChunkScan3dOutputMode_CalibratedC</code>	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.
<code>ChunkScan3dOutputMode_CalibratedC_Linescan</code>	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.
<code>ChunkScan3dOutputMode_RectifiedC</code>	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.
<code>ChunkScan3dOutputMode_RectifiedC_Linescan</code>	Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using <code>Coord3D_C</code> pixels. The B (Y) axis comes from the encoder chunk value.
<code>ChunkScan3dOutputMode_DisparityC</code>	Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.
<code>ChunkScan3dOutputMode_DisparityC_Linescan</code>	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.
<code>NUM_CHUNKSCAN3DOUTPUTMODE</code>	

13.8.1.35 _spinChunkSelectorEnums

```
enum _spinChunkSelectorEnums
```

< 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	

13.8.1.36 _spinChunkSourceIDEnums

```
enum _spinChunkSourceIDEnums
```

< 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	

13.8.1.37 _spinChunkTimerSelectorEnums

```
enum _spinChunkTimerSelectorEnums
```

< 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	

13.8.1.38 _spinChunkTransferStreamIDEnums

```
enum _spinChunkTransferStreamIDEnums
```

< 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	

13.8.1.39 _spinClConfigurationEnums

```
enum _spinClConfigurationEnums
```

< 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

ClConfiguration_Base	Standard base configuration described by the Camera Link standard.
ClConfiguration_Medium	Standard medium configuration described by the Camera Link standard.
ClConfiguration_Full	Standard full configuration described by the Camera Link standard.
ClConfiguration_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.
ClConfiguration_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	

13.8.1.40 _spinCITimeSlotsCountEnums

```
enum _spinCITimeSlotsCountEnums
```

< 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

CITimeSlotsCount_One	One
CITimeSlotsCount_Two	Two
CITimeSlotsCount_Three	Three
NUM_CLTIMESLOTSCOUNT	

13.8.1.41 _spinColorTransformationSelectorEnums

```
enum _spinColorTransformationSelectorEnums
```

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

Enumerator

ColorTransformationSelector_RGBtoRGB	
ColorTransformationSelector_RGBtoYUV	
NUM_COLORTRANSFORMATIONSELECTOR	

13.8.1.42 _spinColorTransformationValueSelectorEnums

```
enum _spinColorTransformationValueSelectorEnums
```

< 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	

13.8.1.43 `_spinCompressionSaturationPriorityEnums`

enum `_spinCompressionSaturationPriorityEnums`

< When FrameRate is enabled, camera drops frames if datarate is saturated. If FrameRate is disabled, camera adjusts the framerate to match the maximum achievable datarate.

Enumerator

CompressionSaturationPriority_DropFrame	Frames which will cause the MaxDatarateThreshold to be exceeded will not be transmitted. Requires FrameRateEnable to be True
CompressionSaturationPriority_ReduceFrameRate	AcquisitionFrameRate is dynamically adjusted to the highest possible value without exceeding the MaxDatarateThreshold.
NUM_COMPRESSIONSATURATIONPRIORITY	

13.8.1.44 `_spinCounterEventActivationEnums`

enum `_spinCounterEventActivationEnums`

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

Enumerator

CounterEventActivation_LevelLow	
CounterEventActivation_LevelHigh	
CounterEventActivation_FallingEdge	
CounterEventActivation_RisingEdge	
CounterEventActivation_AnyEdge	
NUM_COUNTEREVENTACTIVATION	

13.8.1.45 `_spinCounterEventSourceEnums`

enum `_spinCounterEventSourceEnums`

< Selects the event that will increment the counter

Enumerator

CounterEventSource_Off	Off
CounterEventSource_MHzTick	MHzTick
CounterEventSource_Line0	Line0

Enumerator

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	

13.8.1.46 _spinCounterResetActivationEnums

```
enum _spinCounterResetActivationEnums
```

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

Enumerator

CounterResetActivation_LevelLow	
CounterResetActivation_LevelHigh	
CounterResetActivation_FallingEdge	
CounterResetActivation_RisingEdge	
CounterResetActivation_AnyEdge	
NUM_COUNTERRESETACTIVATION	

13.8.1.47 _spinCounterResetSourceEnums

```
enum _spinCounterResetSourceEnums
```

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

Enumerator

CounterResetSource_Off	Off

Enumerator

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	

13.8.1.48 _spinCounterSelectorEnums

```
enum _spinCounterSelectorEnums
```

< Selects which counter to configure

Enumerator

CounterSelector_Counter0	
CounterSelector_Counter1	
NUM_COUNTERSELECTOR	

13.8.1.49 _spinCounterStatusEnums

```
enum _spinCounterStatusEnums
```

< 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	

13.8.1.50 _spinCounterTriggerActivationEnums

```
enum _spinCounterTriggerActivationEnums
```

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

Enumerator

CounterTriggerActivation_LevelLow	
CounterTriggerActivation_LevelHigh	
CounterTriggerActivation_FallingEdge	
CounterTriggerActivation_RisingEdge	
CounterTriggerActivation_AnyEdge	
NUM_COUNTERTRIGGERACTIVATION	

13.8.1.51 _spinCounterTriggerSourceEnums

```
enum _spinCounterTriggerSourceEnums
```

< 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	

13.8.1.52 _spinCxpConnectionTestModeEnums

enum `_spinCxpConnectionTestModeEnums`

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

Enumerator

<code>CxpConnectionTestMode_Off</code>	Off
<code>CxpConnectionTestMode_Mode1</code>	Mode 1
<code>NUM_CXP_CONNECTION_TEST_MODE</code>	

13.8.1.53 _spinCxpLinkConfigurationEnums

enum `_spinCxpLinkConfigurationEnums`

< 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

<code>CxpLinkConfiguration_Auto</code>	Sets Automatic discovery for the Link Configuration.
<code>CxpLinkConfiguration_CXP1_X1</code>	Force the Link to 1 Connection operating at CXP-1 speed (1.25 Gbps).
<code>CxpLinkConfiguration_CXP2_X1</code>	Force the Link to 1 Connection operating at CXP-2 speed (2.50 Gbps).
<code>CxpLinkConfiguration_CXP3_X1</code>	Force the Link to 1 Connection operating at CXP-3 speed (3.125 Gbps).
<code>CxpLinkConfiguration_CXP5_X1</code>	Force the Link to 1 Connection operating at CXP-5 speed (5.00 Gbps).
<code>CxpLinkConfiguration_CXP6_X1</code>	Force the Link to 1 Connection operating at CXP-6 speed (6.25 Gbps).
<code>CxpLinkConfiguration_CXP1_X2</code>	Force the Link to 2 Connections operating at CXP-1 speed (1.25 Gbps).
<code>CxpLinkConfiguration_CXP2_X2</code>	Force the Link to 2 Connections operating at CXP-2 speed (2.50 Gbps).
<code>CxpLinkConfiguration_CXP3_X2</code>	Force the Link to 2 Connections operating at CXP-3 speed (3.125 Gbps).
<code>CxpLinkConfiguration_CXP5_X2</code>	Force the Link to 2 Connections operating at CXP-5 speed (5.00 Gbps).
<code>CxpLinkConfiguration_CXP6_X2</code>	Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).
<code>CxpLinkConfiguration_CXP1_X3</code>	Force the Link to 3 Connections operating at CXP-1 speed (1.25 Gbps).
<code>CxpLinkConfiguration_CXP2_X3</code>	Force the Link to 3 Connections operating at CXP-2 speed (2.50 Gbps).
<code>CxpLinkConfiguration_CXP3_X3</code>	Force the Link to 3 Connections operating at CXP-3 speed (3.125 Gbps).
<code>CxpLinkConfiguration_CXP5_X3</code>	Force the Link to 3 Connections operating at CXP-5 speed (5.00 Gbps).
<code>CxpLinkConfiguration_CXP6_X3</code>	Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).
<code>CxpLinkConfiguration_CXP1_X4</code>	Force the Link to 4 Connections operating at CXP-1 speed (1.25 Gbps).
<code>CxpLinkConfiguration_CXP2_X4</code>	Force the Link to 4 Connections operating at CXP-2 speed (2.50 Gbps).
<code>CxpLinkConfiguration_CXP3_X4</code>	Force the Link to 4 Connections operating at CXP-3 speed (3.125 Gbps).
<code>CxpLinkConfiguration_CXP5_X4</code>	Force the Link to 4 Connections operating at CXP-5 speed (5.00 Gbps).
<code>CxpLinkConfiguration_CXP6_X4</code>	Force the Link to 4 Connections operating at CXP-6 speed (6.25 Gbps).

Enumerator

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	

13.8.1.54 _spinCxpLinkConfigurationPreferredEnums

```
enum _spinCxpLinkConfigurationPreferredEnums
```

< 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).

Enumerator

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	

13.8.1.55 _spinCxpLinkConfigurationStatusEnums

```
enum _spinCxpLinkConfigurationStatusEnums
```

< 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).

Enumerator

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	

13.8.1.56 _spinCxpPoCxpStatusEnums

```
enum _spinCxpPoCxpStatusEnums
```

< 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	

13.8.1.57 _spinDecimationHorizontalModeEnums

```
enum _spinDecimationHorizontalModeEnums
```

< 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	

13.8.1.58 _spinDecimationSelectorEnums

```
enum _spinDecimationSelectorEnums
```

< 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	

13.8.1.59 _spinDecimationVerticalModeEnums

```
enum _spinDecimationVerticalModeEnums
```

< 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	

13.8.1.60 _spinDefectCorrectionModeEnums

```
enum _spinDefectCorrectionModeEnums
```

< 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	

13.8.1.61 _spinDeinterlacingEnums

```
enum _spinDeinterlacingEnums
```

< 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	

13.8.1.62 _spinDeviceCharacterSetEnums

```
enum _spinDeviceCharacterSetEnums
```

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

Enumerator

DeviceCharacterSet_UTF8	
DeviceCharacterSet_ASCII	
NUM_DEVICECHARACTERSET	

13.8.1.63 _spinDeviceClockSelectorEnums

```
enum _spinDeviceClockSelectorEnums
```

< 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	

13.8.1.64 _spinDeviceConnectionStatusEnums

```
enum _spinDeviceConnectionStatusEnums
```

< Indicates the status of the specified Connection.

Enumerator

DeviceConnectionStatus_Active	Connection is in use.
DeviceConnectionStatus_Inactive	Connection is not in use.
NUM_DEVICECONNECTIONSTATUS	

13.8.1.65 `_spinDeviceIndicatorModeEnums`

enum `_spinDeviceIndicatorModeEnums`

< 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	

13.8.1.66 `_spinDeviceLinkHeartbeatModeEnums`

enum `_spinDeviceLinkHeartbeatModeEnums`

< Activate or deactivate the Link's heartbeat.

Enumerator

DeviceLinkHeartbeatMode_On	Enables the Link heartbeat.
DeviceLinkHeartbeatMode_Off	Disables the Link heartbeat.
NUM_DEVICELINKHEARTBEATMODE	

13.8.1.67 `_spinDeviceLinkThroughputLimitModeEnums`

enum `_spinDeviceLinkThroughputLimitModeEnums`

< 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	

13.8.1.68 _spinDevicePowerSupplySelectorEnums

enum `_spinDevicePowerSupplySelectorEnums`

< Selects the power supply source to control or read.

Enumerator

DevicePowerSupplySelector_External	
NUM_DEVICEPOWERSUPPLYSELECTOR	

13.8.1.69 _spinDeviceRegistersEndiannessEnums

enum `_spinDeviceRegistersEndiannessEnums`

< Endianness of the registers of the device.

Enumerator

DeviceRegistersEndianness_Little	
DeviceRegistersEndianness_Big	
NUM_DEVICEREGISTERSENDIANNES	

13.8.1.70 _spinDeviceScanTypeEnums

enum `_spinDeviceScanTypeEnums`

< Scan type of the sensor of the device.

Enumerator

DeviceScanType_Areascan	
NUM_DEVICESCANTYPE	

13.8.1.71 _spinDeviceSerialPortBaudRateEnums

enum `_spinDeviceSerialPortBaudRateEnums`

< 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	

13.8.1.72 _spinDeviceSerialPortSelectorEnums

```
enum _spinDeviceSerialPortSelectorEnums
```

< Selects which serial port of the device to control.

Enumerator

DeviceSerialPortSelector_CameraLink	Serial port associated to the Camera link connection.
NUM_DEVICESERIALPORTSELECTOR	

13.8.1.73 _spinDeviceStreamChannelEndiannessEnums

```
enum _spinDeviceStreamChannelEndiannessEnums
```

< 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	

13.8.1.74 _spinDeviceStreamChannelTypeEnums

```
enum _spinDeviceStreamChannelTypeEnums
```

< Reports the type of the stream channel.

Enumerator

DeviceStreamChannelType_Transmitter	Data stream transmitter channel.
DeviceStreamChannelType_Receiver	Data stream receiver channel.
NUM_DEVICESTREAMCHANNELTYPE	

13.8.1.75 _spinDeviceTapGeometryEnums

enum `_spinDeviceTapGeometryEnums`

< 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

Enumerator

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	

13.8.1.76 `_spinDeviceTemperatureSelectorEnums`

```
enum _spinDeviceTemperatureSelectorEnums
```

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

Enumerator

DeviceTemperatureSelector_Sensor	
NUM_DEVICETEMPERATURESELECTOR	

13.8.1.77 `_spinDeviceTLTypeEnums`

```
enum _spinDeviceTLTypeEnums
```

< Transport Layer type of the device.

Enumerator

DeviceTLType_GigEVision	
DeviceTLType_CameraLink	
DeviceTLType_CameraLinkHS	
DeviceTLType_CoaXPress	
DeviceTLType_USB3Vision	
DeviceTLType_Custom	
NUM_DEVICETLTYPE	

13.8.1.78 _spinDeviceTypeEnums

enum [_spinDeviceTypeEnums](#)

< 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	

13.8.1.79 _spinEncoderModeEnums

enum [_spinEncoderModeEnums](#)

< 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	

13.8.1.80 _spinEncoderOutputModeEnums

enum [_spinEncoderOutputModeEnums](#)

< 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.

Enumerator

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	

13.8.1.81 `_spinEncoderResetActivationEnums`

```
enum _spinEncoderResetActivationEnums
```

< 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	

13.8.1.82 `_spinEncoderResetSourceEnums`

```
enum _spinEncoderResetSourceEnums
```

< 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.

Enumerator

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	

13.8.1.83 _spinEncoderSelectorEnums

```
enum _spinEncoderSelectorEnums
```

< Selects which Encoder to configure.

Enumerator

EncoderSelector_Encoder0	Selects Encoder 0.
EncoderSelector_Encoder1	Selects Encoder 1.
EncoderSelector_Encoder2	Selects Encoder 2.
NUM_ENCODERSELECTOR	

13.8.1.84 `_spinEncoderSourceAEnums`

```
enum _spinEncoderSourceAEnums
```

< 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	

13.8.1.85 `_spinEncoderSourceBEnums`

```
enum _spinEncoderSourceBEnums
```

< 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	

13.8.1.86 `_spinEncoderStatusEnums`

```
enum _spinEncoderStatusEnums
```

< 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	

13.8.1.87 _spinEventNotificationEnums

enum `_spinEventNotificationEnums`

< Enables/Disables the selected event.

Enumerator

EventNotification_On	
EventNotification_Off	
NUM_EVENTNOTIFICATION	

13.8.1.88 _spinEventSelectorEnums

enum `_spinEventSelectorEnums`

< Selects which Event to enable or disable.

Enumerator

EventSelector_Error	
EventSelector_ExposureEnd	
EventSelector_SerialPortReceive	
NUM_EVENTSELECTOR	

13.8.1.89 _spinExposureActiveModeEnums

enum `_spinExposureActiveModeEnums`

< Control sensor active exposure mode.

Enumerator

ExposureActiveMode_Line1	
ExposureActiveMode_AnyPixels	
ExposureActiveMode_AllPixels	
NUM_EXPOSUREACTIVEMODE	

13.8.1.90 `_spinExposureAutoEnums`

enum `_spinExposureAutoEnums`

< Sets the automatic exposure mode

Enumerator

<code>ExposureAuto_Off</code>	Exposure time is manually controlled using <code>ExposureTime</code>
<code>ExposureAuto_Once</code>	Exposure time is adapted once by the device. Once it has converged, it returns to the Off state.
<code>ExposureAuto_Continuous</code>	Exposure time is constantly adapted by the device to maximize the dynamic range.
<code>NUM_EXPOSUREAUTO</code>	

13.8.1.91 `_spinExposureModeEnums`

enum `_spinExposureModeEnums`

< Sets the operation mode of the Exposure.

Enumerator

<code>ExposureMode_Timed</code>	Timed exposure. The exposure time is set using the <code>ExposureTime</code> or <code>ExposureAuto</code> features and the exposure starts with the <code>FrameStart</code> or <code>LineStart</code> .
<code>ExposureMode_TriggerWidth</code>	Uses the width of the current Frame trigger signal pulse to control the exposure time.
<code>NUM_EXPOSUREMODE</code>	

13.8.1.92 `_spinExposureTimeModeEnums`

enum `_spinExposureTimeModeEnums`

< Sets the configuration mode of the `ExposureTime` feature.

Enumerator

<code>ExposureTimeMode_Common</code>	The exposure time is common to all the color components. The common <code>ExposureTime</code> value to use can be set selecting it with <code>ExposureTimeSelector[Common]</code> .
<code>ExposureTimeMode_Individual</code>	The exposure time is individual for each color component. Each individual <code>ExposureTime</code> values to use can be set by selecting them with <code>ExposureTimeSelector</code> .
<code>NUM_EXPOSURETIMEMODE</code>	

13.8.1.93 _spinExposureTimeSelectorEnums

```
enum _spinExposureTimeSelectorEnums
```

< 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	

13.8.1.94 _spinFileOpenModeEnums

```
enum _spinFileOpenModeEnums
```

< 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	

13.8.1.95 _spinFileOperationSelectorEnums

```
enum _spinFileOperationSelectorEnums
```

< 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	

13.8.1.96 _spinFileOperationStatusEnums

```
enum _spinFileOperationStatusEnums
```

< Represents the file operation execution status.

Enumerator

FileOperationStatus_Success	File Operation was successful.
FileOperationStatus_Failure	File Operation failed.
FileOperationStatus_Overflow	An overflow occurred while executing the File Operation.
NUM_FILEOPERATIONSTATUS	

13.8.1.97 _spinFileSelectorEnums

```
enum _spinFileSelectorEnums
```

< 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	

13.8.1.98 _spinGainAutoBalanceEnums

```
enum _spinGainAutoBalanceEnums
```

< 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	

13.8.1.99 _spinGainAutoEnums

```
enum _spinGainAutoEnums
```

< 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	

13.8.1.100 _spinGainSelectorEnums

```
enum _spinGainSelectorEnums
```

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

Enumerator

GainSelector_All	
NUM_GAINSELECTOR	

13.8.1.101 _spinGevCCPEnums

```
enum _spinGevCCPEnums
```

< Controls the device access privilege of an application.

Enumerator

GevCCP_OpenAccess	
GevCCP_ExclusiveAccess	
GevCCP_ControlAccess	
NUM_GEVCCP	

13.8.1.102 _spinGevCurrentPhysicalLinkConfigurationEnums

```
enum _spinGevCurrentPhysicalLinkConfigurationEnums
```

< 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	

13.8.1.103 _spinGevGVCPExtendedStatusCodesSelectorEnums

```
enum _spinGevGVCPExtendedStatusCodesSelectorEnums
```

< 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	

13.8.1.104 _spinGevGVSPExtendedIDModeEnums

```
enum _spinGevGVSPExtendedIDModeEnums
```

< Enables the extended IDs mode.

Enumerator

GevGVSPExtendedIDMode_Off	Off
GevGVSPExtendedIDMode_On	On
NUM_GEVGVSPEXTENDEDIDMODE	

13.8.1.105 _spinGevIEEE1588ClockAccuracyEnums

```
enum _spinGevIEEE1588ClockAccuracyEnums
```

< 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_GEVIEEE1588CLOCKACCURACY	

13.8.1.106 _spinGevIEEE1588ModeEnums

```
enum _spinGevIEEE1588ModeEnums
```

< Provides the mode of the IEEE 1588 clock.

Enumerator

GevIEEE1588Mode_Auto	Automatic
GevIEEE1588Mode_SlaveOnly	Slave Only
NUM_GEVIEEE1588MODE	

13.8.1.107 _spinGevIEEE1588StatusEnums

```
enum _spinGevIEEE1588StatusEnums
```

< Provides the status of the IEEE 1588 clock.

Enumerator

GevIEEE1588Status_Initializing	Initializing
GevIEEE1588Status_Faulty	Faulty
GevIEEE1588Status_Disabled	Disabled

Enumerator

GevIEEE1588Status_Listening	Listening
GevIEEE1588Status_PreMaster	Pre Master
GevIEEE1588Status_Master	Master
GevIEEE1588Status_Passive	Passive
GevIEEE1588Status_Uncalibrated	Uncalibrated
GevIEEE1588Status_Slave	Slave
NUM_GEVIEEE1588STATUS	

13.8.1.108 _spinGevIPConfigurationStatusEnums

```
enum _spinGevIPConfigurationStatusEnums
```

< 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	

13.8.1.109 _spinGevPhysicalLinkConfigurationEnums

```
enum _spinGevPhysicalLinkConfigurationEnums
```

< 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	

13.8.1.110 _spinGevSupportedOptionSelectorEnums

```
enum _spinGevSupportedOptionSelectorEnums
```

< 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	

13.8.1.111 _spinImageComponentSelectorEnums

```
enum _spinImageComponentSelectorEnums
```

< 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.

Enumerator

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	

13.8.1.112 _spinImageCompressionJPEGFormatOptionEnums

```
enum _spinImageCompressionJPEGFormatOptionEnums
```

< 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_Baseline↔ Standard	Indicates this is a baseline sequential (single-scan) DCT-based JPEG.
ImageCompressionJPEGFormatOption_Baseline↔ Optimized	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_IMAGECOMPRESSIONJPEGFORMATOPT↔ ION	

13.8.1.113 _spinImageCompressionModeEnums

```
enum _spinImageCompressionModeEnums
```

<

Enumerator

ImageCompressionMode_Off	
ImageCompressionMode_Lossless	
NUM_IMAGECOMPRESSIONMODE	

13.8.1.114 _spinImageCompressionRateOptionEnums

```
enum _spinImageCompressionRateOptionEnums
```

< 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	

13.8.1.115 _spinLineFormatEnums

```
enum _spinLineFormatEnums
```

< 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	

13.8.1.116 _spinLineInputFilterSelectorEnums

```
enum _spinLineInputFilterSelectorEnums
```

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

Enumerator

LineInputFilterSelector_Deglitch	
LineInputFilterSelector_Debounce	
NUM_LINEINPUTFILTERSELECTOR	

13.8.1.117 _spinLineModeEnums

enum `_spinLineModeEnums`

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

Enumerator

LineMode_Input	
LineMode_Output	
NUM_LINEMODE	

13.8.1.118 _spinLineSelectorEnums

enum `_spinLineSelectorEnums`

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

Enumerator

LineSelector_Line0	
LineSelector_Line1	
LineSelector_Line2	
LineSelector_Line3	
NUM_LINESELECTOR	

13.8.1.119 _spinLineSourceEnums

enum `_spinLineSourceEnums`

< 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	

13.8.1.120 _spinLogicBlockLUTInputActivationEnums

```
enum _spinLogicBlockLUTInputActivationEnums
```

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

Enumerator

LogicBlockLUTInputActivation_LevelLow	
LogicBlockLUTInputActivation_LevelHigh	
LogicBlockLUTInputActivation_FallingEdge	
LogicBlockLUTInputActivation_RisingEdge	
LogicBlockLUTInputActivation_AnyEdge	
NUM_LOGICBLOCKLUTINPUTACTIVATION	

13.8.1.121 _spinLogicBlockLUTInputSelectorEnums

```
enum _spinLogicBlockLUTInputSelectorEnums
```

< Controls which LogicBlockLUT Input Source & Activation to access.

Enumerator

LogicBlockLUTInputSelector_Input0	
LogicBlockLUTInputSelector_Input1	
LogicBlockLUTInputSelector_Input2	
LogicBlockLUTInputSelector_Input3	
NUM_LOGICBLOCKLUTINPUTSELECTOR	

13.8.1.122 _spinLogicBlockLUTInputSourceEnums

enum [_spinLogicBlockLUTInputSourceEnums](#)

< 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	

13.8.1.123 _spinLogicBlockLUTSelectorEnums

enum [_spinLogicBlockLUTSelectorEnums](#)

< Selects which LogicBlock LUT to configure

Enumerator

LogicBlockLUTSelector_Value	
LogicBlockLUTSelector_Enable	
NUM_LOGICBLOCKLUTSELECTOR	

13.8.1.124 _spinLogicBlockSelectorEnums

```
enum _spinLogicBlockSelectorEnums
```

< Selects which LogicBlock to configure

Enumerator

LogicBlockSelector_LogicBlock0	
LogicBlockSelector_LogicBlock1	
NUM_LOGICBLOCKSELECTOR	

13.8.1.125 _spinLUTSelectorEnums

```
enum _spinLUTSelectorEnums
```

The enum definitions for camera nodes.

< 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	

13.8.1.126 _spinPixelColorFilterEnums

```
enum _spinPixelColorFilterEnums
```

< 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.

Enumerator

PixelColorFilter_BayerGB	Bayer Green Blue filter.
PixelColorFilter_BayerGR	Bayer Green Red filter.
PixelColorFilter_BayerBG	Bayer Blue Green filter.
NUM_PIXELCOLORFILTER	

13.8.1.127 _spinPixelFormatEnums

```
enum _spinPixelFormatEnums
```

< 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	

Enumerator

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_Mono16s	Monochrome 16-bit signed
PixelFormat_Mono32f	Monochrome 32-bit float
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_RGB16s	Red-Green-Blue 16-bit signed
PixelFormat_RGB32f	Red-Green-Blue 32-bit float
PixelFormat_RGB16_Planar	Red-Green-Blue 16-bit planar

Enumerator

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_RGBa32f	Red-Green-Blue-alpha 32-bit float
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

Enumerator

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

Enumerator

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

Enumerator

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_LLCMono8	Lossless Compression Monochrome 8-bit
PixelFormat_LLCBayerRG8	Lossless Compression Bayer Red Green filter 8-bit
PixelFormat_JPEGMono8	JPEG Monochrome 8-bit
PixelFormat_JPEGColor8	JPEG Color 8-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.
PixelFormat_GR12	Green-Red (single) channel from Bayer pattern 12-bit.
PixelFormat_GB12	Green-Blue (single) channel from Bayer pattern 12-bit.
UNKNOWN_PIXELFORMAT	
NUM_PIXELFORMAT	

13.8.1.128 _spinPixelFormatInfoSelectorEnums

```
enum _spinPixelFormatInfoSelectorEnums
```

< 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_Mono16s	Monochrome 16-bit signed
PixelFormatInfoSelector_Mono32f	Monochrome 32-bit float
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

Enumerator

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_RGB16s	Red-Green-Blue 16-bit signed
PixelFormatInfoSelector_RGB32f	Red-Green-Blue 32-bit float
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_RGBa32f	Red-Green-Blue-alpha 32-bit float
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

Enumerator

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_SCF1BWBG8	Sparse Color Filter #1 White-Blue-White-Green 8-bit
PixelFormatInfoSelector_SCF1BWBG10	Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked

Enumerator

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

Enumerator

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_CbYY↔ CrYY	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_CbY↔ CrY	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_Cb↔ YCrY	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_CbY↔ CrY	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_Cb↔ YCrY	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_CbYY↔ CrYY	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_CbY↔ CrY	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_422_10p	YCbCr 4:2:2 10-bit packed BT.709

Enumerator

PixelFormatInfoSelector_YCbCr709_422_10p_Cb↔YCrY	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_CbY↔CrY	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_Cb↔YCrY	YCbCr 4:2:2 12-bit packed BT.709
PixelFormatInfoSelector_YUV8_UYV	YUV 4:4:4 8-bit
PixelFormatInfoSelector_YUV411_8_YYYYYY	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
PixelFormatInfoSelector_LLCMono8	Lossless Compression Monochrome 8-bit
PixelFormatInfoSelector_LLCBayerRG8	Lossless Compression Bayer Red Green filter 8-bit
PixelFormatInfoSelector_JPEGMono8	JPEG Monochrome 8-bit
PixelFormatInfoSelector_JPEGColor8	JPEG Color 8-bit
NUM_PIXELFORMATINFOSELECTOR	

13.8.1.129 _spinPixelFormatSizeEnums

```
enum _spinPixelFormatSizeEnums
```

< 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.

Enumerator

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	

13.8.1.130 _spinRegionDestinationEnums

```
enum _spinRegionDestinationEnums
```

< 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	

13.8.1.131 _spinRegionModeEnums

```
enum _spinRegionModeEnums
```

< 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	

13.8.1.132 _spinRegionSelectorEnums

```
enum _spinRegionSelectorEnums
```

< 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	

13.8.1.133 _spinRgbTransformLightSourceEnums

enum `_spinRgbTransformLightSourceEnums`

< 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 ColorTransformationValueSelector and ColorTransformationValue controls.
NUM_RGBTRANSFORMLIGHTSOURCE	

13.8.1.134 _spinScan3dCoordinateReferenceSelectorEnums

enum `_spinScan3dCoordinateReferenceSelectorEnums`

< 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	

13.8.1.135 _spinScan3dCoordinateSelectorEnums

```
enum _spinScan3dCoordinateSelectorEnums
```

< 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	

13.8.1.136 _spinScan3dCoordinateSystemEnums

```
enum _spinScan3dCoordinateSystemEnums
```

< 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	

13.8.1.137 _spinScan3dCoordinateSystemReferenceEnums

```
enum _spinScan3dCoordinateSystemReferenceEnums
```

< 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	

13.8.1.138 _spinScan3dCoordinateTransformSelectorEnums

```
enum _spinScan3dCoordinateTransformSelectorEnums
```

< 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	

13.8.1.139 _spinScan3dDistanceUnitEnums

```
enum _spinScan3dDistanceUnitEnums
```

< 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	

13.8.1.140 _spinScan3dOutputModeEnums

```
enum _spinScan3dOutputModeEnums
```

< 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	

13.8.1.141 _spinSensorDigitizationTapsEnums

enum `_spinSensorDigitizationTapsEnums`

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

Enumerator

SensorDigitizationTaps_One	1 tap.

Enumerator

SensorDigitizationTaps_Two	2 taps.
SensorDigitizationTaps_Three	3 taps.
SensorDigitizationTaps_Four	4 taps.
SensorDigitizationTaps_Eight	8 taps.
SensorDigitizationTaps_Ten	10 taps.
NUM_SENSORDIGITIZATIONTAPS	

13.8.1.142 _spinSensorShutterModeEnums

```
enum _spinSensorShutterModeEnums
```

< 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	

13.8.1.143 _spinSensorTapsEnums

```
enum _spinSensorTapsEnums
```

< 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	

13.8.1.144 _spinSequencerConfigurationModeEnums

```
enum _spinSequencerConfigurationModeEnums
```

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

Enumerator

SequencerConfigurationMode_Off	
SequencerConfigurationMode_On	
NUM_SEQUENCERCONFIGURATIONMODE	

13.8.1.145 _spinSequencerConfigurationValidEnums

```
enum _spinSequencerConfigurationValidEnums
```

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

Enumerator

SequencerConfigurationValid_No	
SequencerConfigurationValid_Yes	
NUM_SEQUENCERCONFIGURATIONVALID	

13.8.1.146 _spinSequencerModeEnums

```
enum _spinSequencerModeEnums
```

< Controls whether or not a sequencer is active.

Enumerator

SequencerMode_Off	
SequencerMode_On	
NUM_SEQUENCERMODE	

13.8.1.147 _spinSequencerSetValidEnums

```
enum _spinSequencerSetValidEnums
```

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

Enumerator

SequencerSetValid_No	
SequencerSetValid_Yes	
NUM_SEQUENCERSETVALID	

13.8.1.148 _spinSequencerTriggerActivationEnums

```
enum _spinSequencerTriggerActivationEnums
```

< Specifies the activation mode of the sequencer trigger.

Enumerator

SequencerTriggerActivation_RisingEdge	
SequencerTriggerActivation_FallingEdge	
SequencerTriggerActivation_AnyEdge	
SequencerTriggerActivation_LevelHigh	
SequencerTriggerActivation_LevelLow	
NUM_SEQUENCERTRIGGERACTIVATION	

13.8.1.149 _spinSequencerTriggerSourceEnums

```
enum _spinSequencerTriggerSourceEnums
```

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

Enumerator

SequencerTriggerSource_Off	
SequencerTriggerSource_FrameStart	
NUM_SEQUENCERTRIGGERSOURCE	

13.8.1.150 _spinSerialPortBaudRateEnums

```
enum _spinSerialPortBaudRateEnums
```

< 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	

13.8.1.151 _spinSerialPortParityEnums

```
enum _spinSerialPortParityEnums
```

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

Enumerator

SerialPortParity_None	
SerialPortParity_Odd	
SerialPortParity_Even	
SerialPortParity_Mark	
SerialPortParity_Space	
NUM_SERIALPORTPARITY	

13.8.1.152 _spinSerialPortSelectorEnums

```
enum _spinSerialPortSelectorEnums
```

< Selects which serial port of the device to control.

Enumerator

SerialPortSelector_SerialPort0	
NUM_SERIALPORTSELECTOR	

13.8.1.153 _spinSerialPortSourceEnums

```
enum _spinSerialPortSourceEnums
```

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

Enumerator

SerialPortSource_Line0	
SerialPortSource_Line1	
SerialPortSource_Line2	
SerialPortSource_Line3	
SerialPortSource_Off	
NUM_SERIALPORTSOURCE	

13.8.1.154 _spinSerialPortStopBitsEnums

```
enum _spinSerialPortStopBitsEnums
```

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

Enumerator

SerialPortStopBits_Bits1	
SerialPortStopBits_Bits1AndAHalf	
SerialPortStopBits_Bits2	
NUM_SERIALPORTSTOPBITS	

13.8.1.155 _spinSoftwareSignalSelectorEnums

```
enum _spinSoftwareSignalSelectorEnums
```

< 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	

13.8.1.156 _spinSourceSelectorEnums

```
enum _spinSourceSelectorEnums
```

< 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	

13.8.1.157 _spinTestPatternEnums

```
enum _spinTestPatternEnums
```

< 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	

13.8.1.158 _spinTestPatternGeneratorSelectorEnums

```
enum _spinTestPatternGeneratorSelectorEnums
```

< 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	

13.8.1.159 _spinTimerSelectorEnums

```
enum _spinTimerSelectorEnums
```

< 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	

13.8.1.160 _spinTimerStatusEnums

```
enum _spinTimerStatusEnums
```

< 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	

13.8.1.161 _spinTimerTriggerActivationEnums

```
enum _spinTimerTriggerActivationEnums
```

< 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	

13.8.1.162 _spinTimerTriggerSourceEnums

```
enum _spinTimerTriggerSourceEnums
```

< 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.

Enumerator

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	

13.8.1.163 `_spinTransferComponentSelectorEnums`

```
enum _spinTransferComponentSelectorEnums
```

< 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	

13.8.1.164 `_spinTransferControlModeEnums`

```
enum _spinTransferControlModeEnums
```

< 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	

13.8.1.165 _spinTransferOperationModeEnums

```
enum _spinTransferOperationModeEnums
```

< 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	

13.8.1.166 _spinTransferQueueModeEnums

```
enum _spinTransferQueueModeEnums
```

< Specifies the operation mode of the transfer queue.

Enumerator

TransferQueueMode_FirstInFirstOut	Blocks first In are transferred Out first.
NUM_TRANSFERQUEUEMODE	

13.8.1.167 _spinTransferSelectorEnums

```
enum _spinTransferSelectorEnums
```

< 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	

13.8.1.168 _spinTransferStatusSelectorEnums

```
enum _spinTransferStatusSelectorEnums
```

< 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	

13.8.1.169 _spinTransferTriggerActivationEnums

```
enum _spinTransferTriggerActivationEnums
```

< 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	

13.8.1.170 _spinTransferTriggerModeEnums

```
enum _spinTransferTriggerModeEnums
```

< Controls if the selected trigger is active.

Enumerator

TransferTriggerMode_Off	Disables the selected trigger.
TransferTriggerMode_On	Enable the selected trigger.
NUM_TRANSFERTRIGGERMODE	

13.8.1.171 _spinTransferTriggerSelectorEnums

```
enum _spinTransferTriggerSelectorEnums
```

< 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	

13.8.1.172 _spinTransferTriggerSourceEnums

```
enum _spinTransferTriggerSourceEnums
```

< 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.

Enumerator

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	

13.8.1.173 _spinTriggerActivationEnums

```
enum _spinTriggerActivationEnums
```

< Specifies the activation mode of the trigger.

Enumerator

TriggerActivation_LevelLow	
TriggerActivation_LevelHigh	
TriggerActivation_FallingEdge	
TriggerActivation_RisingEdge	
TriggerActivation_AnyEdge	
NUM_TRIGGERACTIVATION	

13.8.1.174 _spinTriggerModeEnums

enum `_spinTriggerModeEnums`

< Controls whether or not trigger is active.

Enumerator

TriggerMode_Off	
TriggerMode_On	
NUM_TRIGGERMODE	

13.8.1.175 _spinTriggerOverlapEnums

enum `_spinTriggerOverlapEnums`

< Specifies the overlap mode of the trigger.

Enumerator

TriggerOverlap_Off	
TriggerOverlap_ReadOut	
TriggerOverlap_PreviousFrame	
NUM_TRIGGEROVERLAP	

13.8.1.176 _spinTriggerSelectorEnums

enum `_spinTriggerSelectorEnums`

< Selects the type of trigger to configure.

Enumerator

TriggerSelector_AcquisitionStart	
TriggerSelector_FrameStart	
TriggerSelector_FrameBurstStart	
NUM_TRIGGERSELECTOR	

13.8.1.177 _spinTriggerSourceEnums

```
enum _spinTriggerSourceEnums
```

< 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	

13.8.1.178 _spinUserOutputSelectorEnums

```
enum _spinUserOutputSelectorEnums
```

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

Enumerator

UserOutputSelector_UserOutput0	
UserOutputSelector_UserOutput1	
UserOutputSelector_UserOutput2	
UserOutputSelector_UserOutput3	
NUM_USEROUTPUTSELECTOR	

13.8.1.179 _spinUserSetDefaultEnums

```
enum _spinUserSetDefaultEnums
```

< 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	

13.8.1.180 _spinUserSetSelectorEnums

```
enum _spinUserSetSelectorEnums
```

< 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	

13.8.1.181 _spinWhiteClipSelectorEnums

```
enum _spinWhiteClipSelectorEnums
```

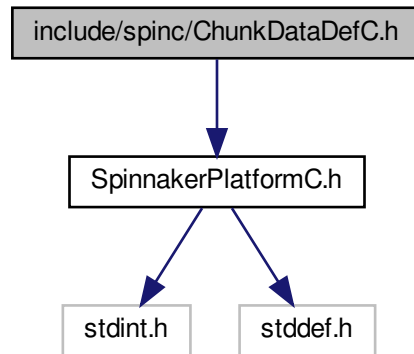
< Selects which White Clip to control.

Enumerator

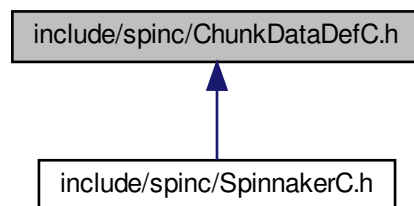
WhiteClipSelector_All	White Clip will be applied to all channels or taps.
WhiteClipSelector_Red	White Clip will be applied to the red channel.
WhiteClipSelector_Green	White Clip will be applied to the green channel.
WhiteClipSelector_Blue	White Clip will be applied to the blue channel.
WhiteClipSelector_Y	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	

13.9 include/spinc/ChunkDataDefC.h File Reference

Include dependency graph for ChunkDataDefC.h:



This graph shows which files directly or indirectly include this file:



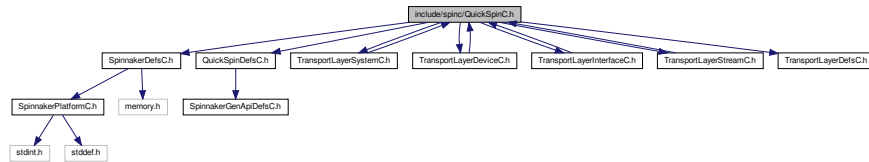
Data Structures

- [struct `\_spinChunkData`](#)

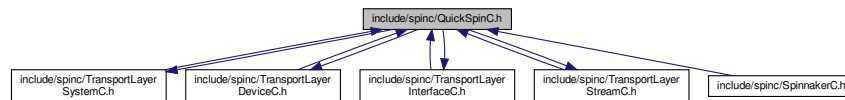
The type of information that can be obtained from image chunk data.

13.10 include/spinc/QuickSpinC.h File Reference

Include dependency graph for QuickSpinC.h:



This graph shows which files directly or indirectly include this file:



Functions

- [SPINNAKERC_API quickSpinInit](#) ([spinCamera](#) hCamera, quickSpin *pQuickSpin)
- [SPINNAKERC_API quickSpinInitEx](#) ([spinCamera](#) hCamera, quickSpin *pQuickSpin, quickSpinTLDevice *pQuickSpinTLDevice, quickSpinTLStream *pQuickSpinTLStream)
- [SPINNAKERC_API quickSpinTLDeviceInit](#) ([spinCamera](#) hCamera, quickSpinTLDevice *pQuickSpinTLDevice)
- [SPINNAKERC_API quickSpinTLStreamInit](#) ([spinCamera](#) hCamera, quickSpinTLStream *pQuickSpinTLStream)
- [SPINNAKERC_API quickSpinTLInterfaceInit](#) ([spinInterface](#) hInterface, quickSpinTLInterface *pQuickSpinTLInterface)
- [SPINNAKERC_API quickSpinTLSystemInit](#) ([spinSystem](#) hSystem, quickSpinTLSystem *pQuickSpinTLSystem)

13.10.1 Function Documentation

13.10.1.1 quickSpinInit()

```

SPINNAKERC_API quickSpinInit (
    spinCamera hCamera,
    quickSpin * pQuickSpin )
  
```

13.10.1.2 quickSpinInitEx()

```
SPINNAKERC_API quickSpinInitEx (
    spinCamera hCamera,
    quickSpin * pQuickSpin,
    quickSpinTLDevice * pQuickSpinTLDevice,
    quickSpinTLStream * pQuickSpinTLStream )
```

13.10.1.3 quickSpinTLDeviceInit()

```
SPINNAKERC_API quickSpinTLDeviceInit (
    spinCamera hCamera,
    quickSpinTLDevice * pQuickSpinTLDevice )
```

13.10.1.4 quickSpinTLInterfaceInit()

```
SPINNAKERC_API quickSpinTLInterfaceInit (
    spinInterface hInterface,
    quickSpinTLInterface * pQuickSpinTLInterface )
```

13.10.1.5 quickSpinTLStreamInit()

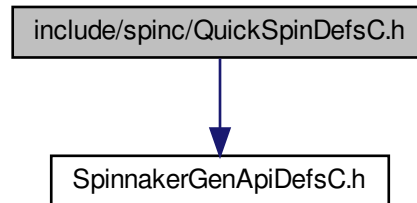
```
SPINNAKERC_API quickSpinTLStreamInit (
    spinCamera hCamera,
    quickSpinTLStream * pQuickSpinTLStream )
```

13.10.1.6 quickSpinTLSystemInit()

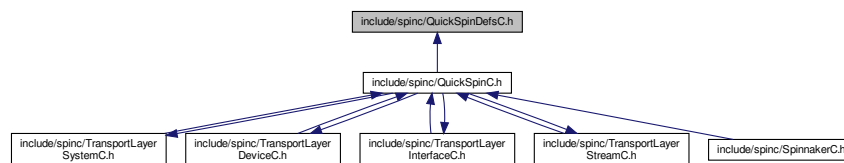
```
SPINNAKERC_API quickSpinTLSystemInit (
    spinSystem hSystem,
    quickSpinTLSystem * pQuickSpinTLSystem )
```

13.11 include/spinc/QuickSpinDefsC.h File Reference

Include dependency graph for QuickSpinDefsC.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [_quickSpin](#)

Typedefs

- typedef [spinNodeHandle](#) [quickSpinStringNode](#)
- typedef [spinNodeHandle](#) [quickSpinIntegerNode](#)
- typedef [spinNodeHandle](#) [quickSpinFloatNode](#)
- typedef [spinNodeHandle](#) [quickSpinBooleanNode](#)
- typedef [spinNodeHandle](#) [quickSpinEnumerationNode](#)
- typedef [spinNodeHandle](#) [quickSpinCommandNode](#)
- typedef [spinNodeHandle](#) [quickSpinRegisterNode](#)

13.11.1 Typedef Documentation

13.11.1.1 quickSpinBooleanNode

```
typedef spinNodeHandle quickSpinBooleanNode
```

13.11.1.2 quickSpinCommandNode

```
typedef spinNodeHandle quickSpinCommandNode
```

13.11.1.3 quickSpinEnumerationNode

```
typedef spinNodeHandle quickSpinEnumerationNode
```

13.11.1.4 quickSpinFloatNode

```
typedef spinNodeHandle quickSpinFloatNode
```

13.11.1.5 quickSpinIntegerNode

```
typedef spinNodeHandle quickSpinIntegerNode
```

13.11.1.6 quickSpinRegisterNode

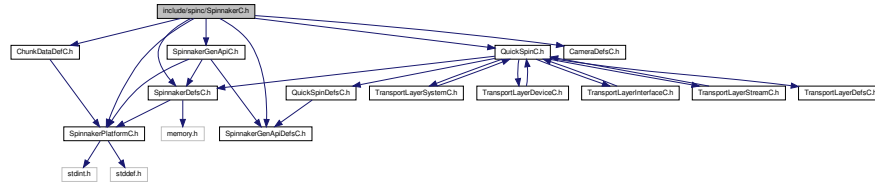
```
typedef spinNodeHandle quickSpinRegisterNode
```

13.11.1.7 quickSpinStringNode

```
typedef spinNodeHandle quickSpinStringNode
```

13.12 include/spinc/SpinnakerC.h File Reference

Include dependency graph for SpinnakerC.h:



Functions

- [SPINNAKERC_API spinErrorGetLast](#) (spinError *pError)
Retrieves the error code of the last error.
- [SPINNAKERC_API spinErrorGetLastMessage](#) (char *pBuf, size_t *pBufLen)
Retrieves the error message of the last error.
- [SPINNAKERC_API spinErrorGetLastBuildDate](#) (char *pBuf, size_t *pBufLen)
Retrieves the build date of the last error.
- [SPINNAKERC_API spinErrorGetLastBuildTime](#) (char *pBuf, size_t *pBufLen)
Retrieves the build time of the last error.
- [SPINNAKERC_API spinErrorGetLastFileName](#) (char *pBuf, size_t *pBufLen)
Retrieves the filename of the last error.
- [SPINNAKERC_API spinErrorGetLastFullMessage](#) (char *pBuf, size_t *pBufLen)
Retrieves the full error message of the last error.
- [SPINNAKERC_API spinErrorGetLastFunctionName](#) (char *pBuf, size_t *pBufLen)
Retrieves the function name of the last error.
- [SPINNAKERC_API spinErrorGetLastLineNumber](#) (int64_t *pLineNum)
Retrieves the line number of the last error.
- [SPINNAKERC_API spinSystemGetInstance](#) (spinSystem *phSystem)
Retrieves an instance of the system object; the system is a singleton, so there will only ever be one instance; system instance must be destroyed by calling spinSystemReleaseInstance.
- [SPINNAKERC_API spinSystemReleaseInstance](#) (spinSystem hSystem)
Releases the system; make sure handle is cleaned up properly by setting it to NULL after system is released; the handle can only be used again after calling spinSystemGetInstance.
- [SPINNAKERC_API spinSystemGetInterfaces](#) (spinSystem hSystem, spinInterfaceList hInterfaceList)
Retrieves a list of detected (and enumerable) interfaces on the system; interface lists must be created and destroyed.
- [SPINNAKERC_API spinSystemGetCameras](#) (spinSystem hSystem, spinCameraList hCameraList)
Retrieves a list of detected (and enumerable) cameras on the system; camera lists must be created and destroyed.
- [SPINNAKERC_API spinSystemGetCamerasEx](#) (spinSystem hSystem, bool8_t bUpdateInterfaces, bool8_t bUpdateCameras, spinCameraList hCameraList)
Retrieves a list of detected (and enumerable) cameras on the system; manually set whether to update the current interface and camera lists; camera lists must be created and destroyed.
- [SPINNAKERC_API spinSystemSetLoggingLevel](#) (spinSystem hSystem, spinnakerLogLevel logLevel)
Sets the logging level for all logging events on the system.
- [SPINNAKERC_API spinSystemGetLoggingLevel](#) (spinSystem hSystem, spinnakerLogLevel *pLogLevel)
Retrieves the logging level for all logging events on the system.
- [SPINNAKERC_API spinSystemRegisterLogEventHandler](#) (spinSystem hSystem, spinLogEventHandler hLogEventHandler)
Registers a log event handler for all logging events on the system.

- Registers a logging event handler to the system (event handlers registered in this way must be unregistered)*

 - [SPINNAKERC_API spinSystemUnregisterLogEventHandler](#) ([spinSystem](#) hSystem, [spinLogEventHandler](#) hLogEventHandler)

Unregisters a selected logging event handler from the system.
- [SPINNAKERC_API spinSystemUnregisterAllLogEventHandlers](#) ([spinSystem](#) hSystem)

Unregisters all logging event handlers from the system.
- [SPINNAKERC_API spinSystemIsInUse](#) ([spinSystem](#) hSystem, [bool8_t](#) *pbIsInUse)

Checks whether a system is currently in use.
- [SPINNAKERC_API spinSystemRegisterDeviceArrivalEventHandler](#) ([spinSystem](#) hSystem, [spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)

Registers a device arrival event handler to every interface on the system (event handlers registered this way must be unregistered)
- [SPINNAKERC_API spinSystemRegisterDeviceRemovalEventHandler](#) ([spinSystem](#) hSystem, [spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)

Registers a device removal event handler to the system to every interface on the system (event handlers registered this way must be unregistered)
- [SPINNAKERC_API spinSystemUnregisterDeviceArrivalEventHandler](#) ([spinSystem](#) hSystem, [spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)

Unregisters a device arrival event handler from the system.
- [SPINNAKERC_API spinSystemUnregisterDeviceRemovalEventHandler](#) ([spinSystem](#) hSystem, [spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)

Unregisters a device removal event handler from the system.
- [SPINNAKERC_API spinSystemRegisterInterfaceEventHandler](#) ([spinSystem](#) hSystem, [spinInterfaceEventHandler](#) hInterfaceEventHandler)

Registers an interface event handler (device arrival and device removal) to every interface on the system (interface events registered this way must be unregistered) If new interfaces are detected by the system after [spinSystemRegisterInterfaceEventHandler\(\)](#) is called, those interfaces will be automatically registered with this event.
- [SPINNAKERC_API spinSystemUnregisterInterfaceEventHandler](#) ([spinSystem](#) hSystem, [spinInterfaceEventHandler](#) hInterfaceEventHandler)

Unregisters an interface event handler from the system.
- [SPINNAKERC_API spinSystemUpdateCameras](#) ([spinSystem](#) hSystem, [bool8_t](#) *pbChanged)

Updates the list of cameras on the system, informing whether there has been any changes.
- [SPINNAKERC_API spinSystemUpdateCamerasEx](#) ([spinSystem](#) hSystem, [bool8_t](#) bUpdateInterfaces, [bool8_t](#) *pbChanged)

Updates the list of cameras on the system, informing whether there has been any changes; manually set whether to update the current interface lists.
- [SPINNAKERC_API spinSystemSendActionCommand](#) ([spinSystem](#) hSystem, [size_t](#) iDeviceKey, [size_t](#) iGroupKey, [size_t](#) iGroupMask, [size_t](#) iActionTime, [size_t](#) *piResultSize, [actionCommandResult](#) results[])

Broadcast an Action Command to all devices on system.
- [SPINNAKERC_API spinSystemGetLibraryVersion](#) ([spinSystem](#) hSystem, [spinLibraryVersion](#) *hLibraryVersion)

Get current library version of Spinnaker.
- [SPINNAKERC_API spinSystemGetTLNodeMap](#) ([spinSystem](#) hSystem, [spinNodeMapHandle](#) *phNodeMap)

Retrieves the transport layer nodemap from the system.
- [SPINNAKERC_API spinInterfaceListCreateEmpty](#) ([spinInterfaceList](#) *phInterfaceList)

Creates an empty interface list (interface lists created this way must be destroyed)
- [SPINNAKERC_API spinInterfaceListDestroy](#) ([spinInterfaceList](#) hInterfaceList)

Destroys an interface list.
- [SPINNAKERC_API spinInterfaceListGetSize](#) ([spinInterfaceList](#) hInterfaceList, [size_t](#) *pSize)

Retrieves the number of interfaces in an interface list.
- [SPINNAKERC_API spinInterfaceListGet](#) ([spinInterfaceList](#) hInterfaceList, [size_t](#) index, [spinInterface](#) *phInterface)

Retrieves an interface from an interface list using an index (interfaces retrieved this way must be released)

- [SPINNAKERC_API spinInterfaceListClear](#) ([spinInterfaceList](#) hInterfaceList)
Clears an interface list.
- [SPINNAKERC_API spinCameraListCreateEmpty](#) ([spinCameraList](#) *phCameraList)
Creates an empty camera list (camera lists created this way must be destroyed)
- [SPINNAKERC_API spinCameraListDestroy](#) ([spinCameraList](#) hCameraList)
Destroys a camera list.
- [SPINNAKERC_API spinCameraListGetSize](#) ([spinCameraList](#) hCameraList, [size_t](#) *pSize)
Retrieves the number of cameras on a camera list.
- [SPINNAKERC_API spinCameraListGet](#) ([spinCameraList](#) hCameraList, [size_t](#) index, [spinCamera](#) *phCamera)
Retrieves a camera from a camera list using an index.
- [SPINNAKERC_API spinCameraListClear](#) ([spinCameraList](#) hCameraList)
Clears a camera list.
- [SPINNAKERC_API spinCameraListRemove](#) ([spinCameraList](#) hCameraList, [size_t](#) index)
Removes a camera from a camera list using its index.
- [SPINNAKERC_API spinCameraListAppend](#) ([spinCameraList](#) hCameraListBase, [spinCameraList](#) hCameraListToAppend)
Appends all the cameras from one camera list to another.
- [SPINNAKERC_API spinCameraListGetBySerial](#) ([spinCameraList](#) hCameraList, [const char](#) *pSerial, [spinCamera](#) *phCamera)
Retrieves a camera from a camera list using its serial number.
- [SPINNAKERC_API spinCameraListRemoveBySerial](#) ([spinCameraList](#) hCameraList, [const char](#) *pSerial)
Removes a camera from a camera list using its serial number.
- [SPINNAKERC_API spinImageListCreateEmpty](#) ([spinImageList](#) *phImageList)
Creates an empty image list (image lists created this way must be destroyed)
- [SPINNAKERC_API spinImageListDestroy](#) ([spinImageList](#) hImageList)
Destroys a image list.
- [SPINNAKERC_API spinImageListGetSize](#) ([spinImageList](#) hImageList, [size_t](#) *pSize)
Retrieves the number of images in an image list.
- [SPINNAKERC_API spinImageListGet](#) ([spinImageList](#) hImageList, [size_t](#) index, [spinImage](#) *phImage)
Retrieves a image from a image list using an index.
- [SPINNAKERC_API spinImageListClear](#) ([spinImageList](#) hImageList)
Clears a image list.
- [SPINNAKERC_API spinImageListRemove](#) ([spinImageList](#) hImageList, [size_t](#) index)
Removes a image from a image list using its index.
- [SPINNAKERC_API spinImageListAppend](#) ([spinImageList](#) hImageListBase, [spinImageList](#) hImageListToAppend)
Appends all the images from one image list to another.
- [SPINNAKERC_API spinImageListGetByPixelFormat](#) ([spinImageList](#) hImageList, [spinPixelFormatEnums](#) pixelFormat, [spinImage](#) *phImage)
Retrieves a image from a image list given its pixel format.
- [SPINNAKERC_API spinImageListRemoveByPixelFormat](#) ([spinImageList](#) hImageList, [spinPixelFormatEnums](#) pixelFormat)
Removes a image from a image list using its pixel format.
- [SPINNAKERC_API spinImageListRelease](#) ([spinImageList](#) hImageList)
- [SPINNAKERC_API spinImageListSave](#) ([spinImageList](#) hImageList, [const char](#) *fileName)
Saves an image list as an object to a file.
- [SPINNAKERC_API spinImageListLoad](#) ([spinImageList](#) *phImageList, [const char](#) *fileName)
Creates an image list object from file.
- [SPINNAKERC_API spinInterfaceUpdateCameras](#) ([spinInterface](#) hInterface, [bool8_t](#) *pbChanged)
Checks whether any cameras have been connected or disconnected on an interface.

- [SPINNAKERC_API spinInterfaceGetCameras](#) ([spinInterface](#) hInterface, [spinCameraList](#) hCameraList)
Retrieves a camera list from an interface; camera lists must be created and destroy.
- [SPINNAKERC_API spinInterfaceGetCamerasEx](#) ([spinInterface](#) hInterface, [bool8_t](#) bUpdateCameras, [spinCameraList](#) hCameraList)
Retrieves a camera list from an interface; manually set whether to update the cameras; camera lists must be created and destroyed.
- [SPINNAKERC_API spinInterfaceGetTLNodeMap](#) ([spinInterface](#) hInterface, [spinNodeMapHandle](#) *phNode↔Map)
Retrieves the transport layer nodemap from an interface.
- [SPINNAKERC_API spinInterfaceRegisterDeviceArrivalEventHandler](#) ([spinInterface](#) hInterface, [spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)
Registers a device arrival event handler on an interface (event handlers registered in this way must be unregistered)
- [SPINNAKERC_API spinInterfaceRegisterDeviceRemovalEventHandler](#) ([spinInterface](#) hInterface, [spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)
Registers a device removal event handler on an interface (event handlers registered in this way must be unregistered)
- [SPINNAKERC_API spinInterfaceUnregisterDeviceArrivalEventHandler](#) ([spinInterface](#) hInterface, [spinDeviceArrivalEventHandler](#) hDeviceArrivalEventHandler)
Unregisters a device arrival event handler from an interface.
- [SPINNAKERC_API spinInterfaceUnregisterDeviceRemovalEventHandler](#) ([spinInterface](#) hInterface, [spinDeviceRemovalEventHandler](#) hDeviceRemovalEventHandler)
Unregisters a device removal event handler from an interface.
- [SPINNAKERC_API spinInterfaceRegisterInterfaceEventHandler](#) ([spinInterface](#) hInterface, [spinInterfaceEventHandler](#) hInterfaceEventHandler)
Registers an interface event handler (both device arrival and device removal) on an interface.
- [SPINNAKERC_API spinInterfaceUnregisterInterfaceEventHandler](#) ([spinInterface](#) hInterface, [spinInterfaceEventHandler](#) hInterfaceEventHandler)
Unregisters an interface event handler from an interface.
- [SPINNAKERC_API spinInterfaceRelease](#) ([spinInterface](#) hInterface)
Releases an interface.
- [SPINNAKERC_API spinInterfaceIsInUse](#) ([spinInterface](#) hInterface, [bool8_t](#) *pbIsInUse)
Checks whether an interface is in use.
- [SPINNAKERC_API spinInterfaceSendActionCommand](#) ([spinInterface](#) hInterface, [size_t](#) iDeviceKey, [size_t](#)↔iGroupKey, [size_t](#) iGroupMask, [size_t](#) iActionTime, [size_t](#) *piResultSize, [actionCommandResult](#) results[])
Broadcast an Action Command to all devices on interface.
- [SPINNAKERC_API spinCameraInit](#) ([spinCamera](#) hCamera)
Initializes a camera, allowing for much more interaction.
- [SPINNAKERC_API spinCameraDeInit](#) ([spinCamera](#) hCamera)
Deinitializes a camera, greatly reducing functionality.
- [SPINNAKERC_API spinCameraGetNodeMap](#) ([spinCamera](#) hCamera, [spinNodeMapHandle](#) *phNodeMap)
Retrieves the GenICam nodemap from a camera.
- [SPINNAKERC_API spinCameraGetTLDeviceNodeMap](#) ([spinCamera](#) hCamera, [spinNodeMapHandle](#) *ph↔NodeMap)
Retrieves the transport layer device nodemap from a camera.
- [SPINNAKERC_API spinCameraGetTLStreamNodeMap](#) ([spinCamera](#) hCamera, [spinNodeMapHandle](#) *ph↔NodeMap)
Retrieves the transport layer stream nodemap from a camera.
- [SPINNAKERC_API spinCameraGetAccessMode](#) ([spinCamera](#) hCamera, [spinAccessMode](#) *pAccessMode)
Retrieves the access mode of a camera (as an enum, spinAccessMode)
- [SPINNAKERC_API spinCameraReadPort](#) ([spinCamera](#) hCamera, [uint64_t](#) iAddress, [void](#) *pBuffer, [size_t](#) iSize)
- [SPINNAKERC_API spinCameraWritePort](#) ([spinCamera](#) hCamera, [uint64_t](#) iAddress, [void](#) *pBuffer, [size_t](#) iSize)
- [SPINNAKERC_API spinCameraBeginAcquisition](#) ([spinCamera](#) hCamera)

- Has a camera start acquiring images.*
- [SPINNAKERC_API spinCameraEndAcquisition](#) ([spinCamera](#) hCamera)
- Has a camera stop acquiring images.*
- [SPINNAKERC_API spinCameraGetNextImage](#) ([spinCamera](#) hCamera, [spinImage](#) *pImage)
- Retrieves an image from a camera.*
- [SPINNAKERC_API spinCameraGetNextImageEx](#) ([spinCamera](#) hCamera, uint64_t grabTimeout, [spinImage](#) *pImage)
- Retrieves an image from a camera; manually set the timeout in milliseconds.*
- [SPINNAKERC_API spinCameraGetNextImageSync](#) ([spinCamera](#) hCamera, uint64_t grabTimeout, [spinImageList](#) *pImageList)
- If a camera supports one or more streams, this function gets one image from each of the camera's streams, and returns the images in a list.*
- [SPINNAKERC_API spinCameraGetUniqueID](#) ([spinCamera](#) hCamera, char *pBuf, size_t *pBufLen)
- Retrieves a unique identifier for a camera.*
- [SPINNAKERC_API spinCamerasStreaming](#) ([spinCamera](#) hCamera, [bool8_t](#) *pbIsStreaming)
- Checks whether a camera is currently acquiring images.*
- [SPINNAKERC_API spinCameraGetGuiXml](#) ([spinCamera](#) hCamera, char *pBuf, size_t *pBufLen)
- Retrieves the GUI XML from a camera.*
- [SPINNAKERC_API spinCameraRegisterDeviceEventHandler](#) ([spinCamera](#) hCamera, [spinDeviceEventHandler](#) hDeviceEventHandler)
- Registers a universal device event handler (every device event type) to a camera.*
- [SPINNAKERC_API spinCameraRegisterDeviceEventHandlerEx](#) ([spinCamera](#) hCamera, [spinDeviceEventHandler](#) hDeviceEventHandler, const char *pName)
- Registers a specific device event handler (only one device event type) to a camera.*
- [SPINNAKERC_API spinCameraUnregisterDeviceEventHandler](#) ([spinCamera](#) hCamera, [spinDeviceEventHandler](#) hDeviceEventHandler)
- Unregisters a device event handler from a camera.*
- [SPINNAKERC_API spinCameraRegisterImageEventHandler](#) ([spinCamera](#) hCamera, [spinImageEventHandler](#) hImageEventHandler)
- Registers an image event handler to a camera.*
- [SPINNAKERC_API spinCameraRegisterImageEventHandlerEx](#) ([spinCamera](#) hCamera, [spinImageEventHandler](#) hImageEventHandler, uint64_t streamIndex)
- Registers an image event handler to a camera Registers a specific stream handler for the camera given a stream index.*
- [SPINNAKERC_API spinCameraUnregisterImageEventHandler](#) ([spinCamera](#) hCamera, [spinImageEventHandler](#) hImageEventHandler)
- Unregisters an image event handler from a camera.*
- [SPINNAKERC_API spinCameraRegisterImageListEventHandler](#) ([spinCamera](#) hCamera, [spinImageListEventHandler](#) hImageListEventHandler)
- Registers an image list event handler to a camera.*
- [SPINNAKERC_API spinCameraUnregisterImageListEventHandler](#) ([spinCamera](#) hCamera, [spinImageListEventHandler](#) hImageListEventHandler)
- Unregisters an image list event handler from a camera.*
- [SPINNAKERC_API spinCameraRelease](#) ([spinCamera](#) hCamera)
- Releases a camera.*
- [SPINNAKERC_API spinCamerasValid](#) ([spinCamera](#) hCamera, [bool8_t](#) *pbValid)
- Checks whether a camera is still valid for use.*
- [SPINNAKERC_API spinCamerasInitialized](#) ([spinCamera](#) hCamera, [bool8_t](#) *pbInit)
- Checks whether a camera is currently initialized.*
- [SPINNAKERC_API spinCameraDiscoverMaxPacketSize](#) ([spinCamera](#) hCamera, unsigned int *pMax↔PacketSize)
- Returns the largest packet size that can be safely used on the interface that device is connected to.*
- [SPINNAKERC_API spinCameraForceIP](#) ()

- Forces the camera to be on the same subnet as its corresponding interface.*

 - [SPINNAKERC_API spinImageCreateEmpty](#) ([spinImage](#) *phImage)

Creates an empty image; images created this way must be destroyed.
- [SPINNAKERC_API spinImageCreate](#) ([spinImage](#) hSrcImage, [spinImage](#) *phDestImage)

Creates an image from another; images created this way must be destroyed.
- [SPINNAKERC_API spinImageCreateEx](#) ([spinImage](#) *phImage, [size_t](#) width, [size_t](#) height, [size_t](#) offsetX, [size_t](#) offsetY, [spinPixelFormatEnums](#) pixelFormat, void *pData)

Creates an image with some set properties; images created this way must be destroyed.
- [SPINNAKERC_API spinImageCreateEx2](#) ([spinImage](#) *phImage, [size_t](#) width, [size_t](#) height, [size_t](#) offsetX, [size_t](#) offsetY, [spinPixelFormatEnums](#) pixelFormat, void *pData, [spinTLPayloadType](#) dataPayloadType, [size_t](#) dataSize)

Creates an image with some set properties; images created this way must be destroyed.
- [SPINNAKERC_API spinImageDestroy](#) ([spinImage](#) hImage)

Destroys an image.
- [SPINNAKERC_API spinImageGetColorProcessing](#) ([spinImage](#) hImage, [spinColorProcessingAlgorithm](#) *pAlgorithm)

Retrieves the color processing algorithm of a specific image.
- [SPINNAKERC_API spinImageReset](#) ([spinImage](#) hImage, [size_t](#) width, [size_t](#) height, [size_t](#) offsetX, [size_t](#) offsetY, [spinPixelFormatEnums](#) pixelFormat)

Resets an image with some set properties.
- [SPINNAKERC_API spinImageResetEx](#) ([spinImage](#) hImage, [size_t](#) width, [size_t](#) height, [size_t](#) offsetX, [size_t](#) offsetY, [spinPixelFormatEnums](#) pixelFormat, void *pData)

Resets an image with some set properties and image data.
- [SPINNAKERC_API spinImageGetID](#) ([spinImage](#) hImage, [uint64_t](#) *pId)

Retrieves the ID of an image.
- [SPINNAKERC_API spinImageGetData](#) ([spinImage](#) hImage, void **ppData)

Retrieves the image data of an image.
- [SPINNAKERC_API spinImageGetPrivateData](#) ([spinImage](#) hImage, void **ppData)

Retrieves the private data of an image.
- [SPINNAKERC_API spinImageGetBufferSize](#) ([spinImage](#) hImage, [size_t](#) *pSize)

Retrieves the buffer size of an image.
- [SPINNAKERC_API spinImageDeepCopy](#) ([spinImage](#) hSrcImage, [spinImage](#) hDestImage)

Creates a deep copy of an image (the destination image must be created as an empty image prior to the deep copy)
- [SPINNAKERC_API spinImageGetWidth](#) ([spinImage](#) hImage, [size_t](#) *pWidth)

Retrieves the width of an image.
- [SPINNAKERC_API spinImageGetHeight](#) ([spinImage](#) hImage, [size_t](#) *pHeight)

Retrieves the height of an image.
- [SPINNAKERC_API spinImageGetOffsetX](#) ([spinImage](#) hImage, [size_t](#) *pOffsetX)

Retrieves the offset of an image along its X axis.
- [SPINNAKERC_API spinImageGetOffsetY](#) ([spinImage](#) hImage, [size_t](#) *pOffsetY)

Retrieves the offset of an image along its Y axis.
- [SPINNAKERC_API spinImageGetPaddingX](#) ([spinImage](#) hImage, [size_t](#) *pPaddingX)

Retrieves the padding of an image along its X axis.
- [SPINNAKERC_API spinImageGetPaddingY](#) ([spinImage](#) hImage, [size_t](#) *pPaddingY)

Retrieves the padding of an image along its Y axis.
- [SPINNAKERC_API spinImageGetFrameID](#) ([spinImage](#) hImage, [uint64_t](#) *pFrameID)

Retrieves the frame ID of an image.
- [SPINNAKERC_API spinImageGetTimeStamp](#) ([spinImage](#) hImage, [uint64_t](#) *pTimeStamp)

Retrieves the timestamp of an image.
- [SPINNAKERC_API spinImageGetPayloadType](#) ([spinImage](#) hImage, [size_t](#) *pPayloadType)

Retrieves the payload type of an image (as an enum, [spinPayloadTypeIn folds](#))

- [SPINNAKERC_API spinImageGetTLPayloadType](#) ([spinImage](#) hImage, spinTLPayloadType *pPayloadType)
Retrieves the transport layer payload type of an image (as an enum, spinPayloadTypeInfolds)
- [SPINNAKERC_API spinImageGetPixelFormat](#) ([spinImage](#) hImage, spinPixelFormatEnums *pPixelFormat)
Retrieves the pixel format of an image (as an enum, spinPixelFormatEnums)
- [SPINNAKERC_API spinImageGetTLPixelFormat](#) ([spinImage](#) hImage, uint64_t *pPixelFormat)
Retrieves the transport layer pixel format of an image (as an unsigned integer)
- [SPINNAKERC_API spinImageGetTLPixelFormatNamespace](#) ([spinImage](#) hImage, spinTLPixelFormatNamespace *pPixelFormatNamespace)
Retrieves the transport layer pixel format namespace of an image (as an enum, spinPixelFormatNamespaceID)
- [SPINNAKERC_API spinImageGetPixelFormatName](#) ([spinImage](#) hImage, char *pBuf, size_t *pBufLen)
Retrieves the pixel format of an image (as a symbolic)
- [SPINNAKERC_API spinImageIsIncomplete](#) ([spinImage](#) hImage, bool8_t *pbIsIncomplete)
Checks whether an image is incomplete.
- [SPINNAKERC_API spinImageGetValidPayloadSize](#) ([spinImage](#) hImage, size_t *pSize)
Retrieves the valid payload size of an image.
- [SPINNAKERC_API spinImageSave](#) ([spinImage](#) hImage, const char *pFilename, spinImageFileFormat format)
Saves an image using a specified file format (using an enum, spinImageFileFormat)
- [SPINNAKERC_API spinImageSaveFromExt](#) ([spinImage](#) hImage, const char *pFilename)
Saves an image using a specified file format (using the extension of the filename)
- [SPINNAKERC_API spinImageSavePng](#) ([spinImage](#) hImage, const char *pFilename, const spinPNGOption *pOption)
Saves an image as a PNG image.
- [SPINNAKERC_API spinImageSavePpm](#) ([spinImage](#) hImage, const char *pFilename, const spinPPMOption *pOption)
Saves an image as a PPM image.
- [SPINNAKERC_API spinImageSavePgm](#) ([spinImage](#) hImage, const char *pFilename, const spinPGMOption *pOption)
Saves an image as an PGM image.
- [SPINNAKERC_API spinImageSaveTiff](#) ([spinImage](#) hImage, const char *pFilename, const spinTIFFOption *pOption)
Saves an image as a TIFF image.
- [SPINNAKERC_API spinImageSaveJpeg](#) ([spinImage](#) hImage, const char *pFilename, const spinJPEGOption *pOption)
Saves an image as a JPEG image.
- [SPINNAKERC_API spinImageSaveJpg2](#) ([spinImage](#) hImage, const char *pFilename, const spinJPG2Option *pOption)
Saves an image as a JPEG 2000 image.
- [SPINNAKERC_API spinImageSaveBmp](#) ([spinImage](#) hImage, const char *pFilename, const spinBMPOption *pOption)
Saves an image as a BMP image.
- [SPINNAKERC_API spinImageGetChunkLayoutID](#) ([spinImage](#) hImage, uint64_t *pId)
Retrieves the chunk layout ID of an image.
- [SPINNAKERC_API spinImageCalculateStatistics](#) ([spinImage](#) hImage, const [spinImageStatistics](#) hStatistics)
Calculates the image statistics of an image.
- [SPINNAKERC_API spinImageGetStatus](#) ([spinImage](#) hImage, spinImageStatus *pStatus)
Retrieves the image status of an image.
- [SPINNAKERC_API spinImageGetStatusDescription](#) (spinImageStatus status, char *pBuf, size_t *pBufLen)
Retrieves the description of image status.
- [SPINNAKERC_API spinImageRelease](#) ([spinImage](#) hImage)
Releases an image.
- [SPINNAKERC_API spinImageHasCRC](#) ([spinImage](#) hImage, bool8_t *pbHasCRC)

- Checks whether an image has CRC.*

 - [SPINNAKERC_API spinImageCheckCRC](#) ([spinImage](#) hImage, [bool8_t](#) *pbCheckCRC)

Checks whether the CRC of an image is correct.
- [SPINNAKERC_API spinImageGetBitsPerPixel](#) ([spinImage](#) hImage, [size_t](#) *pBitsPerPixel)

Retrieves the number of bits per pixel of an image.
- [SPINNAKERC_API spinImageGetSize](#) ([spinImage](#) hImage, [size_t](#) *pImageSize)

Retrieves the size of an image.
- [SPINNAKERC_API spinImageGetStride](#) ([spinImage](#) hImage, [size_t](#) *pStride)

Retrieves the stride of an image.
- [SPINNAKERC_API spinImageProcessorCreate](#) ([spinImageProcessor](#) *phImageProcessor)

Creates an image processor.
- [SPINNAKERC_API spinImageProcessorDestroy](#) ([spinImageProcessor](#) hImageProcessor)

Destroys a image list.
- [SPINNAKERC_API spinImageProcessorSetColorProcessing](#) ([spinImageProcessor](#) hImageProcessor, [spinColorProcessingAlgorithm](#) colorAlgorithm)

Sets the color processing algorithm used at the time of the [spinImageProcessorConvert\(\)](#) call, therefore the most recent execution of this function will take precedence.
- [SPINNAKERC_API spinImageProcessorGetColorProcessing](#) ([spinImageProcessor](#) hImageProcessor, [spinColorProcessingAlgorithm](#) *pColorAlgorithm)

Gets the default color processing algorithm.
- [SPINNAKERC_API spinImageProcessorSetNumDecompressionThreads](#) ([spinImageProcessor](#) hImageProcessor, unsigned int numThreads)

Sets the default number of threads used for image decompression during [spinImageProcessorConvert\(\)](#).
- [SPINNAKERC_API spinImageProcessorGetNumDecompressionThreads](#) ([spinImageProcessor](#) hImageProcessor, unsigned int *pNumThreads)

Gets the number of threads used for image decompression during [spinImageProcessorConvert\(\)](#).
- [SPINNAKERC_API spinImageProcessorConvert](#) ([spinImageProcessor](#) hImageProcessor, [spinImage](#) hSrcImage, [spinImage](#) hDestImage, [spinPixelFormatEnums](#) destFormat)

Converts the source image buffer to the specified destination pixel format and stores the result in the destination image.
- [SPINNAKERC_API spinImageProcessorConvertImageList](#) ([spinImageProcessor](#) hImageProcessor, [spinImageList](#) hSrcImageList, [spinImage](#) hDestImage, [spinPixelFormatEnums](#) destFormat)

Converts the source list of image buffers to the specified output pixel format and returns the result in a new image.
- [SPINNAKERC_API spinImageProcessorApplyGamma](#) ([spinImageProcessor](#) hImageProcessor, [spinImage](#) hSrcImage, [spinImage](#) hDestImage, float gamma, [bool8_t](#) applyGammaInverse)

Applies gamma correction to the source image and stores the result in the destination image.
- [SPINNAKERC_API spinDeviceEventHandlerCreate](#) ([spinDeviceEventHandler](#) *phDeviceEventHandler, [spinDeviceEventFunction](#) pFunction, void *pUserData)

Creates a device event handler.
- [SPINNAKERC_API spinDeviceEventHandlerDestroy](#) ([spinDeviceEventHandler](#) hDeviceEventHandler)

Destroys a device event handler.
- [SPINNAKERC_API spinImageEventHandlerCreate](#) ([spinImageEventHandler](#) *phImageEventHandler, [spinImageEventFunction](#) pFunction, void *pUserData)

Creates an image event handler.
- [SPINNAKERC_API spinImageEventHandlerDestroy](#) ([spinImageEventHandler](#) hImageEventHandler)

Destroys an image event handler.
- [SPINNAKERC_API spinImageListEventHandlerCreate](#) ([spinImageListEventHandler](#) *phImageEventHandler, [spinImageListEventFunction](#) pFunction, void *pUserData)

Creates an image list event handler.
- [SPINNAKERC_API spinImageListEventHandlerDestroy](#) ([spinImageListEventHandler](#) hImageListEventHandler)

Destroys an image list event handler.

- [SPINNAKERC_API spinDeviceArrivalEventHandlerCreate](#) ([spinDeviceArrivalEventHandler](#) *phDevice↔
ArrivalEventHandler, [spinArrivalEventFunction](#) pFunction, void *pUserData)
Creates a device arrival event handler.
- [SPINNAKERC_API spinDeviceArrivalEventHandlerDestroy](#) ([spinDeviceArrivalEventHandler](#) hDevice↔
ArrivalEventHandler)
Destroys a device arrival event handler.
- [SPINNAKERC_API spinDeviceRemovalEventHandlerCreate](#) ([spinDeviceRemovalEventHandler](#) *phDevice↔
DeviceRemovalEventHandler, [spinRemovalEventFunction](#) pFunction, void *pUserData)
Creates a device removal event handler.
- [SPINNAKERC_API spinDeviceRemovalEventHandlerDestroy](#) ([spinDeviceRemovalEventHandler](#) hDevice↔
RemovalEventHandler)
Destroys a device removal event handler.
- [SPINNAKERC_API spinInterfaceEventHandlerCreate](#) ([spinInterfaceEventHandler](#) *phInterfaceEvent↔
Handler, [spinArrivalEventFunction](#) pArrivalFunction, [spinRemovalEventFunction](#) pRemovalFunction, void
*pUserData)
Creates an interface event handler (both device arrival and device removal)
- [SPINNAKERC_API spinInterfaceEventHandlerDestroy](#) ([spinInterfaceEventHandler](#) hInterfaceEventHandler)
Destroys an interface event handler (both device arrival and device removal)
- [SPINNAKERC_API spinLogEventHandlerCreate](#) ([spinLogEventHandler](#) *phLogEventHandler, [spinLogEventFunction](#)
pFunction, void *pUserData)
Creates a log event handler.
- [SPINNAKERC_API spinLogEventHandlerDestroy](#) ([spinLogEventHandler](#) hLogEventHandler)
Destroys a log event handler.
- [SPINNAKERC_API spinImageStatisticsCreate](#) ([spinImageStatistics](#) *phStatistics)
Creates an image statistics context.
- [SPINNAKERC_API spinImageStatisticsDestroy](#) ([spinImageStatistics](#) hStatistics)
Destroys an image statistics context.
- [SPINNAKERC_API spinImageStatisticsEnableAll](#) ([spinImageStatistics](#) hStatistics)
Enables all channels of an image statistics context.
- [SPINNAKERC_API spinImageStatisticsDisableAll](#) ([spinImageStatistics](#) hStatistics)
Disables all channels of an image statistics context.
- [SPINNAKERC_API spinImageStatisticsEnableGreyOnly](#) ([spinImageStatistics](#) hStatistics)
Disables all channels of an image statistics context except grey-scale.
- [SPINNAKERC_API spinImageStatisticsEnableRgbOnly](#) ([spinImageStatistics](#) hStatistics)
Disables all channels of an image statistics context except red, blue, and green.
- [SPINNAKERC_API spinImageStatisticsEnableHslOnly](#) ([spinImageStatistics](#) hStatistics)
Disables all channels of an image statistics context except hue, saturation, and lightness.
- [SPINNAKERC_API spinImageStatisticsGetChannelStatus](#) ([spinImageStatistics](#) hStatistics, [spinStatistics](#)↔
Channel channel, [bool8_t](#) *pbEnabled)
Checks whether an image statistics context is enabled.
- [SPINNAKERC_API spinImageStatisticsSetChannelStatus](#) ([spinImageStatistics](#) hStatistics, [spinStatistics](#)↔
Channel channel, [bool8_t](#) bEnable)
Sets the status of an image statistics channel.
- [SPINNAKERC_API spinImageStatisticsGetRange](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#)
channel, unsigned int *pMin, unsigned int *pMax)
Retrieves the range of an image statistics channel.
- [SPINNAKERC_API spinImageStatisticsGetPixelValueRange](#) ([spinImageStatistics](#) hStatistics, [spin](#)↔
StatisticsChannel channel, unsigned int *pMin, unsigned int *pMax)
Retrieves the pixel value range of an image statistics channel.
- [SPINNAKERC_API spinImageStatisticsGetNumPixelValues](#) ([spinImageStatistics](#) hStatistics, [spinStatistics](#)↔
Channel channel, unsigned int *pNumValues)
Retrieves the number of pixel values of an image statistics channel.

- [SPINNAKERC_API spinImageStatisticsGetMean](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#) channel, float *pMean)
Retrieves the mean of pixel values of an image statistics channel.
- [SPINNAKERC_API spinImageStatisticsGetHistogram](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#) channel, int **ppHistogram)
Retrieves a histogram of an image statistics channel.
- [SPINNAKERC_API spinImageStatisticsGetAll](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#) channel, unsigned int *pRangeMin, unsigned int *pRangeMax, unsigned int *pPixelValueMin, unsigned int *pPixelValueMax, unsigned int *pNumPixelValues, float *pPixelValueMean, int **ppHistogram)
Retrieves all available information of an image statistics channel.
- [SPINNAKERC_API spinLogDataGetCategoryName](#) ([spinLogEventData](#) hLogEventData, char *pBuf, size_t *pBufLen)
Retrieves the category name of a log event.
- [SPINNAKERC_API spinLogDataGetPriority](#) ([spinLogEventData](#) hLogEventData, int64_t *pValue)
Retrieves the priority of a log event.
- [SPINNAKERC_API spinLogDataGetPriorityName](#) ([spinLogEventData](#) hLogEventData, char *pBuf, size_t *pBufLen)
Retrieves the priority name of a log event.
- [SPINNAKERC_API spinLogDataGetTimestamp](#) ([spinLogEventData](#) hLogEventData, char *pBuf, size_t *pBufLen)
Retrieves the timestamp of a log event.
- [SPINNAKERC_API spinLogDataGetNDC](#) ([spinLogEventData](#) hLogEventData, char *pBuf, size_t *pBufLen)
Retrieves the NDC of a log event.
- [SPINNAKERC_API spinLogDataGetThreadName](#) ([spinLogEventData](#) hLogEventData, char *pBuf, size_t *pBufLen)
Retrieves the thread name of a log event.
- [SPINNAKERC_API spinLogDataGetLogMessage](#) ([spinLogEventData](#) hLogEventData, char *pBuf, size_t *pBufLen)
Retrieves the log message of a log event.
- [SPINNAKERC_API spinDeviceEventGetId](#) ([spinDeviceEventData](#) hDeviceEventData, uint64_t *pEventId)
Retrieves the event ID of a device event.
- [SPINNAKERC_API spinDeviceEventGetPayloadData](#) ([spinDeviceEventData](#) hDeviceEventData, const uint8_t *pBuf, size_t *pBufSize)
Retrieves the payload data of a device event.
- [SPINNAKERC_API spinDeviceEventGetPayloadDataSize](#) ([spinDeviceEventData](#) hDeviceEventData, size_t *pBufSize)
Retrieves the payload data size of a device event.
- [SPINNAKERC_API spinDeviceEventGetName](#) ([spinDeviceEventData](#) hDeviceEventData, char *pBuf, size_t *pBufLen)
Retrieves the event name of a device event.
- [SPINNAKERC_API spinImageChunkDataGetIntValue](#) ([spinImage](#) hImage, const char *pName, int64_t *pValue)
Retrieves the integer value of a spin image chunk.
- [SPINNAKERC_API spinImageChunkDataGetFloatValue](#) ([spinImage](#) hImage, const char *pName, double *pValue)
Retrieves the float value of a spin image chunk.

13.12.1 Function Documentation

13.12.1.1 spinCameraBeginAcquisition()

```
SPINNAKERC_API spinCameraBeginAcquisition (  
    spinCamera hCamera )
```

Has a camera start acquiring images.

See also

spinError

Parameters

<i>hCamera</i>	The camera to begin acquiring images

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.2 spinCameraDeInit()

```
SPINNAKERC_API spinCameraDeInit (  
    spinCamera hCamera )
```

Deinitializes a camera, greatly reducing functionality.

See also

spinError

Parameters

<i>hCamera</i>	The camera to deinitialize

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.3 spinCameraDiscoverMaxPacketSize()

```
SPINNAKERC_API spinCameraDiscoverMaxPacketSize (  
    spinCamera hCamera,  
    unsigned int * pMaxPacketSize )
```

Returns the largest packet size that can be safely used on the interface that device is connected to.

See also

spinError

Parameters

<i>hCamera</i>	The camera to check
<i>pMaxPacketSize</i>	The maximum packet size returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.4 spinCameraEndAcquisition()

```
SPINNAKERC_API spinCameraEndAcquisition (
    spinCamera hCamera )
```

Has a camera stop acquiring images.

See also

spinError

Parameters

<i>hCamera</i>	The camera to stop acquiring images

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.5 spinCameraForceIP()

```
SPINNAKERC_API spinCameraForceIP ( )
```

Forces the camera to be on the same subnet as its corresponding interface.

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.6 spinCameraGetAccessMode()

```
SPINNAKERC_API spinCameraGetAccessMode (
    spinCamera hCamera,
    spinAccessMode * pAccessMode )
```

Retrieves the access mode of a camera (as an enum, spinAccessMode)

See also

spinError
spinAccessMode

Parameters

<i>hCamera</i>	The camera of the access mode to retrieve
<i>pAccessMode</i>	The access mode enum pointer in which the access mode is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.7 spinCameraGetGuiXml()

```
SPINNAKERC_API spinCameraGetGuiXml (
    spinCamera hCamera,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the GUI XML from a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera of the GUI XML to retrieve
<i>pBuf</i>	The c-string character buffer in which the GUI XML is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.8 spinCameraGetNextImage()

```
SPINNAKERC_API spinCameraGetNextImage (
    spinCamera hCamera,
    spinImage * phImage )
```

Retrieves an image from a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera of the image to retrieve
<i>phImage</i>	The image handle pointer in which the image is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.9 spinCameraGetNextImageEx()

```
SPINNAKERC_API spinCameraGetNextImageEx (
    spinCamera hCamera,
    uint64_t grabTimeout,
    spinImage * phImage )
```

Retrieves an image from a camera; manually set the timeout in milliseconds.

See also

spinError

Parameters

<i>hCamera</i>	The camera of the image to retrieve
<i>grabTimeout</i>	A 64bit value that represents a timeout in milliseconds
<i>phImage</i>	The image handle pointer in which the image is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.10 spinCameraGetNextImageSync()

```
SPINNAKERC_API spinCameraGetNextImageSync (
    spinCamera hCamera,
    uint64_t grabTimeout,
    spinImageList * phImageList )
```

If a camera supports one or more streams, this function gets one image from each of the camera's streams, and returns the images in a list.

This function will block for the specified timeout period until an image arrives on all the streams.

See also

[spinCameraInit\(\)](#)
[spinCameraBeginAcquisition\(\)](#)
[spinCameraEndAcquisition\(\)](#)

Parameters

<i>hCamera</i>	The camera of the image to retrieve
<i>grabTimeout</i>	A 64bit value that represents a timeout in milliseconds

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.11 spinCameraGetNodeMap()

```
SPINNAKERC_API spinCameraGetNodeMap (
    spinCamera hCamera,
    spinNodeMapHandle * phNodeMap )
```

Retrieves the GenICam nodemap from a camera.

See also

[spinError](#)

Parameters

<i>hCamera</i>	The camera from which the nodemap is retrieved
<i>phNodeMap</i>	The nodemap handle pointer in which the nodemap is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.12 spinCameraGetTLDeviceNodeMap()

```
SPINNAKERC_API spinCameraGetTLDeviceNodeMap (
    spinCamera hCamera,
    spinNodeMapHandle * phNodeMap )
```

Retrieves the transport layer device nodemap from a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera from which the transport layer device nodemap is retrieved
<i>phNodeMap</i>	The nodemap handle pointer in which the nodemap is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.13 spinCameraGetTLStreamNodeMap()

```
SPINNAKERC_API spinCameraGetTLStreamNodeMap (
    spinCamera hCamera,
    spinNodeMapHandle * phNodeMap )
```

Retrieves the transport layer stream nodemap from a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera from which the transport layer streaming nodemap is retrieved
<i>phNodeMap</i>	The nodemap handle pointer in which the nodemap is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.14 spinCameraGetUniqueID()

```
SPINNAKERC_API spinCameraGetUniqueID (
    spinCamera hCamera,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves a unique identifier for a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera of the unique identifier
<i>pBuf</i>	The c-string character buffer in which the unique identifier is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.15 spinCameraInit()

```
SPINNAKERC_API spinCameraInit (
    spinCamera hCamera )
```

Initializes a camera, allowing for much more interaction.

See also

spinError

Parameters

<i>hCamera</i>	The camera to initialize

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.16 spinCameraIsInitialized()

```
SPINNAKERC_API spinCameraIsInitialized (
    spinCamera hCamera,
    bool8_t * pbInit )
```

Checks whether a camera is currently initialized.

See also

spinError

Parameters

<i>hCamera</i>	The camera to check
<i>pbInit</i>	The boolean pointer to return whether or not the camera is initialized

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.17 spinCameraIsStreaming()

```
SPINNAKERC_API spinCameraIsStreaming (
    spinCamera hCamera,
    bool8_t * pbIsStreaming )
```

Checks whether a camera is currently acquiring images.

See also

spinError

Parameters

<i>hCamera</i>	The camera to check
<i>pbIsStreaming</i>	The boolean pointer to return whether or not the camera is currently streaming

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.18 spinCameraIsValid()

```
SPINNAKERC_API spinCameraIsValid (
    spinCamera hCamera,
    bool8_t * pbValid )
```

Checks whether a camera is still valid for use.

See also

`spinError`

Parameters

<i>hCamera</i>	The camera to check
<i>pbValid</i>	The boolean pointer to return whether or not the camera is valid

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.19 `spinCameraListAppend()`

```
SPINNAKERC_API spinCameraListAppend (  
    spinCameraList hCameraListBase,  
    spinCameraList hCameraListToAppend )
```

Appends all the cameras from one camera list to another.

See also

`spinError`

Parameters

<i>hCameraListBase</i>	The camera list to receive the other
<i>hCameraListToAppend</i>	The camera list to add to the other

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.20 `spinCameraListClear()`

```
SPINNAKERC_API spinCameraListClear (  
    spinCameraList hCameraList )
```

Clears a camera list.

See also

`spinError`

Parameters

<i>hCameraList</i>	The camera list to clear

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.21 spinCameraListCreateEmpty()

```
SPINNAKERC_API spinCameraListCreateEmpty (  
    spinCameraList * phCameraList )
```

Creates an empty camera list (camera lists created this way must be destroyed)

See also

`spinError`

Parameters

<i>phCameraList</i>	The camera list handle pointer in which the empty camera list is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.22 spinCameraListDestroy()

```
SPINNAKERC_API spinCameraListDestroy (  
    spinCameraList hCameraList )
```

Destroys a camera list.

See also

`spinError`

Parameters

<i>hCameraList</i>	The camera list to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.23 spinCameraListGet()

```
SPINNAKERC_API spinCameraListGet (
    spinCameraList hCameraList,
    size_t index,
    spinCamera * phCamera )
```

Retrieves a camera from a camera list using an index.

This function will return a SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

See also

spinError

Parameters

<i>hCameraList</i>	The camera list of the camera to retrieve
<i>index</i>	The index of the camera
<i>phCamera</i>	The camera handle pointer in which the camera is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.24 spinCameraListGetBySerial()

```
SPINNAKERC_API spinCameraListGetBySerial (
    spinCameraList hCameraList,
    const char * pSerial,
    spinCamera * phCamera )
```

Retrieves a camera from a camera list using its serial number.

This function will return a NULL spinCamera pointer if no matching camera serial is found.

See also

spinError

Parameters

<i>hCameraList</i>	The camera list of the camera to retrieve
<i>serial</i>	The serial number of the camera to retrieve
<i>phCamera</i>	The camera handle pointer in which the camera is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.25 spinCameraListGetSize()

```
SPINNAKERC_API spinCameraListGetSize (  
    spinCameraList hCameraList,  
    size_t * pSize )
```

Retrieves the number of cameras on a camera list.

See also

spinError

Parameters

<i>hCameraList</i>	The camera list where the cameras to be counted are
<i>pSize</i>	The unsigned integer pointer in which the number of cameras is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.26 spinCameraListRemove()

```
SPINNAKERC_API spinCameraListRemove (  
    spinCameraList hCameraList,  
    size_t index )
```

Removes a camera from a camera list using its index.

See also

spinError

Parameters

<i>hCameraList</i>	The camera list of the camera to remove
<i>index</i>	The index of the camera to remove

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.27 spinCameraListRemoveBySerial()

```
SPINNAKERC_API spinCameraListRemoveBySerial (
    spinCameraList hCameraList,
    const char * pSerial )
```

Removes a camera from a camera list using its serial number.

See also

`spinError`

Parameters

<i>hCameraList</i>	The camera list of the camera to remove
<i>pSerial</i>	The serial number of the camera to remove

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.28 spinCameraReadPort()

```
SPINNAKERC_API spinCameraReadPort (
    spinCamera hCamera,
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize )
```

13.12.1.29 spinCameraRegisterDeviceEventHandler()

```
SPINNAKERC_API spinCameraRegisterDeviceEventHandler (
    spinCamera hCamera,
    spinDeviceEventHandler hDeviceEventHandler )
```

Registers a universal device event handler (every device event type) to a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera on which to register the universal device event handler
<i>hDeviceEventHandler</i>	The device event handler to register

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.30 spinCameraRegisterDeviceEventHandlerEx()

```
SPINNAKERC_API spinCameraRegisterDeviceEventHandlerEx (
    spinCamera hCamera,
    spinDeviceEventHandler hDeviceEventHandler,
    const char * pName )
```

Registers a specific device event handler (only one device event type) to a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera on which to register the specific device event handler
<i>hDeviceEventHandler</i>	The device event handler to register
<i>pName</i>	The name of the device event handler to register

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.31 spinCameraRegisterImageEventHandler()

```
SPINNAKERC_API spinCameraRegisterImageEventHandler (
    spinCamera hCamera,
    spinImageEventHandler hImageEventHandler )
```

Registers an image event handler to a camera.

See also

[spinError](#)

Parameters

<i>hCamera</i>	The camera on which to register the image event handler
<i>hImageEventHandler</i>	The image event handler to register

Returns

[spinError](#) The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.32 spinCameraRegisterImageEventHandlerEx()

```
SPINNAKERC_API spinCameraRegisterImageEventHandlerEx (
    spinCamera hCamera,
    spinImageEventHandler hImageEventHandler,
    uint64_t streamIndex )
```

Registers an image event handler to a camera Registers a specific stream handler for the camera given a stream index.

The camera has to be initialized first with a call to [spinCameraInit\(\)](#) before registering handlers for events.

See also

[spinError](#)

Parameters

<i>hCamera</i>	The camera on which to register the image event handler
<i>hImageEventHandler</i>	The image event handler to register
<i>streamIndex</i>	The index of the stream of where this handler will be registered to

Returns

[spinError](#) The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.33 spinCameraRegisterImageListEventHandler()

```
SPINNAKERC_API spinCameraRegisterImageListEventHandler (
    spinCamera hCamera,
    spinImageListEventHandler hImageListEventHandler )
```

Registers an image list event handler to a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera on which to register the image event handler
<i>hImageListEventHandler</i>	The image list event handler to register

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.34 spinCameraRelease()

```
SPINNAKERC_API spinCameraRelease (
    spinCamera hCamera )
```

Releases a camera.

See also

spinError

Parameters

<i>hCamera</i>	The camera to release

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.35 spinCameraUnregisterDeviceEventHandler()

```
SPINNAKERC_API spinCameraUnregisterDeviceEventHandler (
    spinCamera hCamera,
    spinDeviceEventHandler hDeviceEventHandler )
```

Unregisters a device event handler from a camera.

See also

`spinError`

Parameters

<i>hCamera</i>	The camera from which to unregister the device event handler
<i>hDeviceEventHandler</i>	The device event handler to unregister

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.36 `spinCameraUnregisterImageEventHandler()`

```
SPINNAKERC_API spinCameraUnregisterImageEventHandler (
    spinCamera hCamera,
    spinImageEventHandler hImageEventHandler )
```

Unregisters an image event handler from a camera.

See also

`spinError`

Parameters

<i>hCamera</i>	The camera from which to unregister the image event handler
<i>hImageEventHandler</i>	The image event handler to unregister

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.37 `spinCameraUnregisterImageListEventHandler()`

```
SPINNAKERC_API spinCameraUnregisterImageListEventHandler (
    spinCamera hCamera,
    spinImageListEventHandler hImageListEventHandler )
```

Unregisters an image list event handler from a camera.

See also

`spinError`

Parameters

<i>hCamera</i>	The camera from which to unregister the image event handler
<i>hImageEventHandler</i>	The image event handler to unregister

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.38 spinCameraWritePort()

```
SPINNAKERC_API spinCameraWritePort (
    spinCamera hCamera,
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize )
```

13.12.1.39 spinDeviceArrivalEventHandlerCreate()

```
SPINNAKERC_API spinDeviceArrivalEventHandlerCreate (
    spinDeviceArrivalEventHandler * phDeviceArrivalEventHandler,
    spinArrivalEventFunction pFunction,
    void * pUserData )
```

Creates a device arrival event handler.

See also

`spinError`

Parameters

<i>phDeviceArrivalEventHandler</i>	The device arrival event handler pointer in which the device arrival event context is created
<i>pFunction</i>	The function to be called at device event occurrences; signature to match: <code>void(spinArrivalEventFunction)(void pUserData)</code>
<i>pUserData</i>	Properties that can be passed into the event function

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.40 spinDeviceArrivalEventHandlerDestroy()

```
SPINNAKERC_API spinDeviceArrivalEventHandlerDestroy (
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Destroys a device arrival event handler.

See also

spinError

Parameters

<i>hDeviceArrivalEventHandler</i>	The device arrival event handler to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.41 spinDeviceEventGetId()

```
SPINNAKERC_API spinDeviceEventGetId (
    spinDeviceEventData hDeviceEventData,
    uint64_t * pEventId )
```

Retrieves the event ID of a device event.

See also

spinError

Parameters

<i>hDeviceEventData</i>	The log event data received from the log event
<i>pEventId</i>	The unsigned integer pointer in which the event ID is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.42 spinDeviceEventGetName()

```
SPINNAKERC_API spinDeviceEventGetName (
    spinDeviceEventData hDeviceEventData,
```

```
char * pBuf,  
size_t * pBufLen )
```

Retrieves the event name of a device event.

See also

spinError

Parameters

<i>hDeviceEventData</i>	The log event data received from the log event
<i>pBuf</i>	The c-string character buffer in which the name of the device event is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.43 spinDeviceEventGetPayloadData()

```
SPINNAKERC_API spinDeviceEventGetPayloadData (  
    spinDeviceEventData hDeviceEventData,  
    const uint8_t * pBuf,  
    size_t * pBufSize )
```

Retrieves the payload data of a device event.

See also

spinError

Parameters

<i>hDeviceEventData</i>	The log event data received from the log event
<i>pBuf</i>	The unsigned integer pointer in which the event payload is returned
<i>pBufSize</i>	The unsigned integer pointer in which the size of the payload is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.44 spinDeviceEventGetPayloadDataSize()

```
SPINNAKERC_API spinDeviceEventGetPayloadDataSize (
    spinDeviceEventData hDeviceEventData,
    size_t * pBufSize )
```

Retrieves the payload data size of a device event.

See also

spinError

Parameters

<i>hDeviceEventData</i>	The log event data received from the log event
<i>pBufSize</i>	The unsigned integer pointer in which the size of the payload is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.45 spinDeviceEventHandlerCreate()

```
SPINNAKERC_API spinDeviceEventHandlerCreate (
    spinDeviceEventHandler * phDeviceEventHandler,
    spinDeviceEventFunction pFunction,
    void * pUserData )
```

Creates a device event handler.

See also

spinError

Parameters

<i>phDeviceEventHandler</i>	The device event handler pointer in which the device event context is created
<i>pFunction</i>	The function to be called at device event occurrences; signature to match: void(spinDeviceEventFunction)(const spinDeviceEventData hEventData, const char pEventName, void* pUserData)
<i>pUserData</i>	Properties that can be passed into the event function

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.46 spinDeviceEventHandlerDestroy()

```
SPINNAKERC_API spinDeviceEventHandlerDestroy (
    spinDeviceEventHandler hDeviceEventHandler )
```

Destroys a device event handler.

See also

spinError

Parameters

<i>hDeviceEventHandler</i>	The device event handler to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.47 spinDeviceRemovalEventHandlerCreate()

```
SPINNAKERC_API spinDeviceRemovalEventHandlerCreate (
    spinDeviceRemovalEventHandler * phDeviceRemovalEventHandler,
    spinRemovalEventFunction pFunction,
    void * pUserData )
```

Creates a device removal event handler.

See also

spinError

Parameters

<i>phDeviceRemovalEventHandler</i>	The device removal event handler pointer in which the device removal event context is created
<i>pFunction</i>	The function to be called at device event occurrences; signature to match: void(spinRemovalEventFunction)(uint64_t deviceSerialNumber, void pUserData)
<i>pUserData</i>	Properties that can be passed into the event function

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.48 spinDeviceRemovalEventHandlerDestroy()

```
SPINNAKERC_API spinDeviceRemovalEventHandlerDestroy (
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Destroys a device removal event handler.

See also

spinError

Parameters

<i>hDeviceRemovalEventHandler</i>	The device removal event handler to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.49 spinErrorGetLast()

```
SPINNAKERC_API spinErrorGetLast (
    spinError * pError )
```

Retrieves the error code of the last error.

See also

spinError

Parameters

<i>pError</i>	The error enum pointer in which the error message is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.50 spinErrorGetLastBuildDate()

```
SPINNAKERC_API spinErrorGetLastBuildDate (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the build date of the last error.

See also

spinError

Parameters

<i>pBuf</i>	The c-string character buffer in which the build date is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.51 spinErrorGetLastBuildTime()

```
SPINNAKERC_API spinErrorGetLastBuildTime (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the build time of the last error.

See also

spinError

Parameters

<i>pBuf</i>	The c-string character buffer in which the build time is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.52 spinErrorGetLastFileName()

```
SPINNAKERC_API spinErrorGetLastFileName (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the filename of the last error.

See also

spinError

Parameters

<i>pBuf</i>	The c-string character buffer in which the file name is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.53 spinErrorGetLastFullMessage()

```
SPINNAKERC_API spinErrorGetLastFullMessage (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the full error message of the last error.

See also

spinError

Parameters

<i>pBuf</i>	The c-string character buffer in which the full error message is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.54 spinErrorGetLastFunctionName()

```
SPINNAKERC_API spinErrorGetLastFunctionName (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the function name of the last error.

See also

spinError

Parameters

<i>pBuf</i>	The c-string character buffer in which the function name is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.55 spinErrorGetLastLineNumber()

```
SPINNAKERC_API spinErrorGetLastLineNumber (
    int64_t * pLineNum )
```

Retrieves the line number of the last error.

See also

spinError

Parameters

<i>pBuf</i>	The c-string character buffer in which the line number is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.56 spinErrorGetLastMessage()

```
SPINNAKERC_API spinErrorGetLastMessage (
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the error message of the last error.

See also

spinError

Parameters

<i>pBuf</i>	The c-string character buffer in which the error message is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.57 spinImageCalculateStatistics()

```
SPINNAKERC_API spinImageCalculateStatistics (
    spinImage hImage,
    const spinImageStatistics hStatistics )
```

Calculates the image statistics of an image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>hStatistics</i>	The image statistics context in which the calculated statistics are returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.58 spinImageCheckCRC()

```
SPINNAKERC_API spinImageCheckCRC (
    spinImage hImage,
    bool8_t * pbCheckCRC )
```

Checks whether the CRC of an image is correct.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pbCheckCRC</i>	The boolean pointer to return whether the image CRC passes

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.59 spinImageChunkDataGetFloatValue()

```
SPINNAKERC_API spinImageChunkDataGetFloatValue (
    spinImage hImage,
    const char * pName,
    double * pValue )
```

13.12.1.60 spinImageChunkDataGetIntValue()

```
SPINNAKERC_API spinImageChunkDataGetIntValue (
    spinImage hImage,
    const char * pName,
    int64_t * pValue )
```

13.12.1.61 spinImageCreate()

```
SPINNAKERC_API spinImageCreate (
    spinImage hSrcImage,
    spinImage * phDestImage )
```

Creates an image from another; images created this way must be destroyed.

See also

spinError

Parameters

<i>hSrcImage</i>	The image to be copied
<i>phDestImage</i>	The image handle pointer of the image to be created

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.62 spinImageCreateEmpty()

```
SPINNAKERC_API spinImageCreateEmpty (
    spinImage * phImage )
```

Creates an empty image; images created this way must be destroyed.

See also

`spinError`

Parameters

<i>phImage</i>	The image handle pointer in which the empty image is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.63 spinImageCreateEx()

```
SPINNAKERC_API spinImageCreateEx (
    spinImage * phImage,
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    spinPixelFormatEnums pixelFormat,
    void * pData )
```

Creates an image with some set properties; images created this way must be destroyed.

See also

`spinError`

Parameters

<i>phImage</i>	The image handle pointer in which the image is returned
<i>width</i>	The width to set
<i>height</i>	The height to set
<i>offsetX</i>	The offset along the X axis to set
<i>offsetY</i>	The offset along the Y axis to set
<i>pixelFormat</i>	The pixel format to set
<i>pData</i>	The image data to set; can be set to null

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.64 spinImageCreateEx2()

```
SPINNAKERC_API spinImageCreateEx2 (
    spinImage * phImage,
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    spinPixelFormatEnums pixelFormat,
    void * pData,
    spinTLPayloadType dataPayloadType,
    size_t dataSize )
```

Creates an image with some set properties; images created this way must be destroyed.

See also

`spinError`

[spinImageGetTLPayloadType](#)

Parameters

<i>phImage</i>	The image handle pointer in which the image is returned
<i>width</i>	The width to set
<i>height</i>	The height to set
<i>offsetX</i>	The offset along the X axis to set
<i>offsetY</i>	The offset along the Y axis to set
<i>pixelFormat</i>	The pixel format to set
<i>pData</i>	The image data to set; can be set to null
<i>dataPayloadType</i>	The payload type of the data. This value can be retrieved from an existing image by using the spinImageGetTLPayloadType() function call.
<i>dataSize</i>	The size of the provided data in bytes

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.65 spinImageDeepCopy()

```
SPINNAKERC_API spinImageDeepCopy (
    spinImage hSrcImage,
    spinImage hDestImage )
```

Creates a deep copy of an image (the destination image must be created as an empty image prior to the deep copy)

See also

spinError

Parameters

<i>hSrcImage</i>	The image to be copied
<i>hDestImage</i>	The image handle in which the image is copied

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.66 spinImageDestroy()

```
SPINNAKERC_API spinImageDestroy (  
    spinImage hImage )
```

Destroys an image.

See also

spinError

Parameters

<i>hImage</i>	The image to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.67 spinImageEventHandlerCreate()

```
SPINNAKERC_API spinImageEventHandlerCreate (  
    spinImageEventHandler * phImageEventHandler,  
    spinImageEventFunction pFunction,  
    void * pUserData )
```

Creates an image event handler.

See also

spinError

Parameters

<i>phImageEventHandler</i>	The image event handler pointer in which the image event context is created
<i>pFunction</i>	The function to be called at image event occurrences; signature to match: void(spinImageEventFunction)(const spinImage hImage, void pUserData)
<i>pUserData</i>	Properties that can be passed into the event function

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.68 spinImageEventHandlerDestroy()

```
SPINNAKERC_API spinImageEventHandlerDestroy (
    spinImageEventHandler hImageEventHandler )
```

Destroys an image event handler.

See also

spinError

Parameters

<i>hImageEventHandler</i>	The image event handler to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.69 spinImageGetBitsPerPixel()

```
SPINNAKERC_API spinImageGetBitsPerPixel (
    spinImage hImage,
    size_t * pBitsPerPixel )
```

Retrieves the number of bits per pixel of an image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pBitsPerPixel</i>	The unsigned integer pointer in which the number of bits per pixel is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.70 spinImageGetBufferSize()

```
SPINNAKERC_API spinImageGetBufferSize (  
    spinImage hImage,  
    size_t * pSize )
```

Retrieves the buffer size of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of image data buffer to retrieve
<i>pSize</i>	The unsigned integer pointer in which the size of the image data if returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.71 spinImageGetChunkLayoutID()

```
SPINNAKERC_API spinImageGetChunkLayoutID (  
    spinImage hImage,  
    uint64_t * pId )
```

Retrieves the chunk layout ID of an image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pId</i>	The unsigned integer pointer in which the chunk layout ID is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.72 spinImageGetColorProcessing()

```
SPINNAKERC_API spinImageGetColorProcessing (
    spinImage hImage,
    spinColorProcessingAlgorithm * pAlgorithm )
```

Retrieves the color processing algorithm of a specific image.

See also

spinError

Parameters

<i>hImage</i>	The image of the color processing algorithm to retrieve
<i>pAlgorithm</i>	The color processing algorithm pointer in which the color processing algorithm is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.73 spinImageGetData()

```
SPINNAKERC_API spinImageGetData (
    spinImage hImage,
    void ** ppData )
```

Retrieves the image data of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of the image data to retrieve
<i>ppData</i>	The pointer to the void pointer in which the image data is retrieved

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.74 spinImageGetFrameID()

```
SPINNAKERC_API spinImageGetFrameID (
    spinImage hImage,
    uint64_t * pFrameID )
```

Retrieves the frame ID of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of the frame ID to retrieve
<i>pFrameID</i>	The unsigned integer pointer in which the frame ID is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.75 spinImageGetHeight()

```
SPINNAKERC_API spinImageGetHeight (
    spinImage hImage,
    size_t * pHeight )
```

Retrieves the height of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of the height to retrieve
<i>pHeight</i>	The unsigned integer pointer in which the height is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.76 spinImageGetID()

```
SPINNAKERC_API spinImageGetID (
    spinImage hImage,
    uint64_t * pId )
```

Retrieves the ID of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of the ID to retrieve
<i>pId</i>	The unsigned integer pointer in which the ID is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.77 spinImageGetOffsetX()

```
SPINNAKERC_API spinImageGetOffsetX (
    spinImage hImage,
    size_t * pOffsetX )
```

Retrieves the offset of an image along its X axis.

See also

spinError

Parameters

<i>hImage</i>	The image of the offset along the X axis to retrieve
<i>pOffsetX</i>	The unsigned integer pointer in which the offset along the X axis is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.78 spinImageGetOffsetY()

```
SPINNAKERC_API spinImageGetOffsetY (
    spinImage hImage,
    size_t * pOffsetY )
```

Retrieves the offset of an image along its Y axis.

See also

spinError

Parameters

<i>hImage</i>	The image of the offset along the Y axis to retrieve
<i>pOffsetY</i>	The unsigned integer pointer in which the offset along the Y axis is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.79 spinImageGetPaddingX()

```
SPINNAKERC_API spinImageGetPaddingX (
    spinImage hImage,
    size_t * pPaddingX )
```

Retrieves the padding of an image along its X axis.

See also

spinError

Parameters

<i>hImage</i>	The image of the padding along the X axis to retrieve
<i>pPaddingX</i>	The unsigned integer pointer in which the padding along the X axis is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.80 spinImageGetPaddingY()

```
SPINNAKERC_API spinImageGetPaddingY (
    spinImage hImage,
    size_t * pPaddingY )
```

Retrieves the padding of an image along its Y axis.

See also

spinError

Parameters

<i>hImage</i>	The image of the padding along the Y axis to retrieve
<i>pPaddingY</i>	The unsigned integer pointer in which the padding along the Y axis is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.81 spinImageGetPayloadType()

```
SPINNAKERC_API spinImageGetPayloadType (
    spinImage hImage,
    size_t * pPayloadType )
```

Retrieves the payload type of an image (as an enum, spinPayloadTypeInfolds)

See also

spinError

spinPayloadTypeInfolds

Parameters

<i>hImage</i>	The image of the payload type to retrieve
<i>pPayloadType</i>	The payload type enum pointer in which the payload type is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.82 spinImageGetPixelFormat()

```
SPINNAKERC_API spinImageGetPixelFormat (
    spinImage hImage,
    spinPixelFormatEnums * pPixelFormat )
```

Retrieves the pixel format of an image (as an enum, spinPixelFormatEnums)

See also

spinError
spinPixelFormatEnums

Parameters

<i>hImage</i>	The image of the pixel format to retrieve
<i>pPixelFormat</i>	The pixel format enum pointer in which the pixel format is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.83 spinImageGetPixelFormatName()

```
SPINNAKERC_API spinImageGetPixelFormatName (
    spinImage hImage,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the pixel format of an image (as a symbolic)

See also

spinError

Parameters

<i>hImage</i>	The image of the pixel format to retrieve
<i>pBuf</i>	The c-string character buffer in which the pixel format symbolic is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.84 spinImageGetPrivateData()

```
SPINNAKERC_API spinImageGetPrivateData (
    spinImage hImage,
    void ** ppData )
```

Retrieves the private data of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of the private image data to retrieve
<i>ppData</i>	The pointer to the void pointer in which the private image data is retrieved

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.85 spinImageGetSize()

```
SPINNAKERC_API spinImageGetSize (
    spinImage hImage,
    size_t * pImageSize )
```

Retrieves the size of an image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pImageSize</i>	The unsigned integer pointer in which the size of the image is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.86 spinImageGetStatus()

```
SPINNAKERC_API spinImageGetStatus (
    spinImage hImage,
    spinImageStatus * pStatus )
```

Retrieves the image status of an image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pStatus</i>	The status enum pointer in which the image status is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.87 spinImageGetStatusDescription()

```
SPINNAKERC_API spinImageGetStatusDescription (
    spinImageStatus status,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the description of image status.

See also

spinError

Parameters

<i>status</i>	The status enum
<i>pBuf</i>	The c-string character buffer in which the explanation of image status enum is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length; if pBuf is NULL, minimum length of string buffer is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.88 spinImageGetStride()

```
SPINNAKERC_API spinImageGetStride (
    spinImage hImage,
    size_t * pStride )
```

Retrieves the stride of an image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pStride</i>	The unsigned integer pointer in which the stride is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.89 spinImageGetTimeStamp()

```
SPINNAKERC_API spinImageGetTimeStamp (
    spinImage hImage,
    uint64_t * pTimeStamp )
```

Retrieves the timestamp of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of the timestamp to retrieve
<i>pTimeStamp</i>	The unsigned integer pointer om which the timestamp is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.90 spinImageGetTLPayloadType()

```
SPINNAKERC_API spinImageGetTLPayloadType (  
    spinImage hImage,  
    spinTLPayloadType * pPayloadType )
```

Retrieves the transport layer payload type of an image (as an enum, spinPayloadTypeInfolds)

See also

spinError
spinPayloadTypeInfolds

Parameters

<i>hImage</i>	The image of the TL payload type to retrieve
<i>pPayloadType</i>	The payload type enum pointer in which the TL payload type is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.91 spinImageGetTLPixelFormat()

```
SPINNAKERC_API spinImageGetTLPixelFormat (  
    spinImage hImage,  
    uint64_t * pPixelFormat )
```

Retrieves the transport layer pixel format of an image (as an unsigned integer)

See also

spinError

Parameters

<i>hImage</i>	The image of the TL pixel format to retrieve
<i>pPixelFormat</i>	The unsigned integer pointer in which the TL pixel format is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.92 spinImageGetTLPixelFormatNamespace()

```
SPINNAKERC_API spinImageGetTLPixelFormatNamespace (
    spinImage hImage,
    spinTLPixelFormatNamespace * pPixelFormatNamespace )
```

Retrieves the transport layer pixel format namespace of an image (as an enum, spinPixelFormatNamespaceID)

See also

spinError
spinPixelFormatNamespaceID

Parameters

<i>hImage</i>	The image of the TL pixel format namespace to retrieve
<i>pPixelFormatNamespace</i>	The pixel format namespace pointer in which the pixel format namespace is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.93 spinImageGetValidPayloadSize()

```
SPINNAKERC_API spinImageGetValidPayloadSize (
    spinImage hImage,
    size_t * pSize )
```

Retrieves the valid payload size of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of the payload size to retrieve
<i>pSize</i>	The unsigned integer pointer in which the size of the valid payload is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.94 spinImageGetWidth()

```
SPINNAKERC_API spinImageGetWidth (
    spinImage hImage,
    size_t * pWidth )
```

Retrieves the width of an image.

See also

spinError

Parameters

<i>hImage</i>	The image of the width to retrieve
<i>pWidth</i>	The unsigned integer pointer in which the width is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.95 spinImageHasCRC()

```
SPINNAKERC_API spinImageHasCRC (
    spinImage hImage,
    bool8_t * pbHasCRC )
```

Checks whether an image has CRC.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pbHasCRC</i>	The boolean pointer to return whether the image has CRC available

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.96 spinImageIsIncomplete()

```
SPINNAKERC_API spinImageIsIncomplete (
    spinImage hImage,
    bool8_t * pbIsIncomplete )
```

Checks whether an image is incomplete.

See also

spinError

Parameters

<i>hImage</i>	The image to check
<i>pbIsIncomplete</i>	The boolean pointer to return whether or not the image is incomplete

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.97 spinImageListAppend()

```
SPINNAKERC_API spinImageListAppend (
    spinImageList hImageListBase,
    spinImageList hImageListToAppend )
```

Appends all the images from one image list to another.

See also

spinError

Parameters

<i>hImageListBase</i>	The image list to receive the other
<i>hImageListToAppend</i>	The image list to add to the other

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.98 spinImageListClear()

```
SPINNAKERC_API spinImageListClear (
    spinImageList hImageList )
```

Clears a image list.

See also

spinError

Parameters

<i>hImageList</i>	The image list to clear

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.99 spinImageListCreateEmpty()

```
SPINNAKERC_API spinImageListCreateEmpty (
    spinImageList * phImageList )
```

Creates an empty image list (image lists created this way must be destroyed)

See also

spinError

Parameters

<i>phImageList</i>	The image list handle pointer in which the empty image list is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.100 spinImageListDestroy()

```
SPINNAKERC_API spinImageListDestroy (
    spinImageList hImageList )
```

Destroys a image list.

See also

spinError

Parameters

<i>hImageList</i>	The image list to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.101 spinImageListEventHandlerCreate()

```
SPINNAKERC_API spinImageListEventHandlerCreate (
    spinImageListEventHandler * phImageEventHandler,
    spinImageListEventFunction pFunction,
    void * pUserData )
```

Creates an image list event handler.

See also

spinError

Parameters

<i>phImageListEventHandler</i>	The image list event handler pointer in which the image list event context is created
<i>pFunction</i>	The function to be called at image list event occurrences; signature to match: void(spinImageListEventFunction)(const spinListImage hImage, void pUserData)
<i>pUserData</i>	Properties that can be passed into the event function

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.102 spinImageListEventHandlerDestroy()

```
SPINNAKERC_API spinImageListEventHandlerDestroy (
    spinImageListEventHandler hImageListEventHandler )
```

Destroys an image list event handler.

See also

spinError

Parameters

<i>hImageListEventHandler</i>	The image list event handler to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.103 spinImageListGet()

```
SPINNAKERC_API spinImageListGet (
    spinImageList hImageList,
    size_t index,
    spinImage * phImage )
```

Retrieves a image from a image list using an index.

This function will return a SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

See also

spinError

Parameters

<i>hImageList</i>	The image list of the image to retrieve
<i>index</i>	The index of the image
<i>phImage</i>	The image handle pointer in which the image is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.104 spinImageListGetByPixelFormat()

```
SPINNAKERC_API spinImageListGetByPixelFormat (
    spinImageList hImageList,
    spinPixelFormatEnums pixelFormat,
    spinImage * phImage )
```

Retrieves a image from a image list given its pixel format.

This function will return a NULL `spinImage` pointer if no matching image pixel format is found.

See also

`spinError`

Parameters

<i>hImageList</i>	The image list of the image to retrieve
<i>pixelFormat</i>	The pixel format of the image to retrieve
<i>phImage</i>	The image handle pointer in which the image is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.105 spinImageListGetSize()

```
SPINNAKERC_API spinImageListGetSize (
    spinImageList hImageList,
    size_t * pSize )
```

Retrieves the number of images in an image list.

See also

`spinError`

Parameters

<i>hImageList</i>	The image list where the images to be counted are
<i>pSize</i>	The unsigned integer pointer in which the number of images is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.106 spinImageListLoad()

```
SPINNAKERC_API spinImageListLoad (
    spinImageList * phImageList,
    const char * fileName )
```

Creates an image list object from file.

See also

[spinImageListSave\(\)](#)

spinError

Parameters

<i>phImageList</i>	The image list handle pointer in which the empty image list is returned
<i>fileName</i>	Name of the file to load an image object from.

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.107 spinImageListRelease()

```
SPINNAKERC_API spinImageListRelease (
    spinImageList hImageList )
```

13.12.1.108 spinImageListRemove()

```
SPINNAKERC_API spinImageListRemove (
    spinImageList hImageList,
    size_t index )
```

Removes a image from a image list using its index.

See also

spinError

Parameters

<i>hImageList</i>	The image list of the camera to remove
<i>index</i>	The index of the image to remove

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.109 spinImageListRemoveByPixelFormat()

```
SPINNAKERC_API spinImageListRemoveByPixelFormat (
    spinImageList hImageList,
    spinPixelFormatEnums pixelFormat )
```

Removes a image from a image list using its pixel format.

See also

`spinError`

Parameters

<i>hImageList</i>	The image list of the image to remove
<i>pixelFormat</i>	The pixel format of the image to remove

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.110 spinImageListSave()

```
SPINNAKERC_API spinImageListSave (
    spinImageList hImageList,
    const char * fileName )
```

Saves an image list as an object to a file.

See also

[spinImageListLoad\(\)](#)

`spinError`

Parameters

<i>hImageList</i>	The image list of the image to remove
<i>fileName</i>	Name of the file to save the current image list object to. It is recommended to use the file extension 'sil'.

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.111 spinImageProcessorApplyGamma()

```
SPINNAKERC_API spinImageProcessorApplyGamma (
    spinImageProcessor hImageProcessor,
    spinImage hSrcImage,
    spinImage hDestImage,
    float gamma,
    bool8_t applyGammaInverse )
```

Applies gamma correction to the source image and stores the result in the destination image.

Parameters

<i>hImageProcessor</i>	The image processor context
<i>hSrcImage</i>	The source image from which to apply gamma on.
<i>hDestImage</i>	The destination image in which the gamma applied image data will be stored.
<i>gamma</i>	Gamma value to apply. A value between 0.5 and 4 is acceptable. (Default assuming image-to-screen)
<i>applyGammaInverse</i>	Converts a gamma corrected source image back to the original image using the inverse of the gamma value (used for applying screen-to-image gamma)

13.12.1.112 spinImageProcessorConvert()

```
SPINNAKERC_API spinImageProcessorConvert (
    spinImageProcessor hImageProcessor,
    spinImage hSrcImage,
    spinImage hDestImage,
    spinPixelFormatEnums destFormat )
```

Converts the source image buffer to the specified destination pixel format and stores the result in the destination image.

The destination image needs to be configured to have the correct buffer size before calling this function. See [spinImageReset\(\)](#) to setup the correct buffer size according to specified pixel format.

Note that compressed images are decompressed before any further color processing or conversion during this call. Decompression is multi-threaded and defaults to utilizing one less than the number of concurrent threads supported by the system. The default number of decompression threads can be set with [spinImageProcessorSetNumDecompressionThreads\(\)](#).

See also

[spinPixelFormatEnums](#)
[spinImageReset](#)
[spinImageProcessorSetNumDecompressionThreads](#)

Parameters

<i>hImageProcessor</i>	The image processor context
<i>srcImage</i>	The source image from which to convert the image from.
<i>destImage</i>	The destination image in which the converted image data will be stored.
<i>destFormat</i>	Output format of the converted image.

13.12.1.113 spinImageProcessorConvertImageList()

```
SPINNAKERC_API spinImageProcessorConvertImageList (
    spinImageProcessor hImageProcessor,
    spinImageList hSrcImageList,
    spinImage hDestImage,
    spinPixelFormatEnums destFormat )
```

Converts the source list of image buffers to the specified output pixel format and returns the result in a new image.

The conversion could encompasses decompression, interleaving and conversion of image data depending on the source pixel format of images in the source image list. The destination image needs to be configured to have the correct buffer size before calling this function. See [spinImageReset\(\)](#) to setup the correct buffer size according to specified pixel format.

Note that compressed images are decompressed before any further color processing, interleaving or conversion is performed. Decompression is multi-threaded and defaults to utilizing one less than the number of concurrent threads supported by the system. The default number of decompression threads can be set with [SetNumDecompressionThreads\(\)](#).

Note not all the supported image pixel formats described in the class description are supported in this function.

List of supported image pixel formats for the source image list:

- PixelFormat_R12
- PixelFormat_GR12
- PixelFormat_GB12
- PixelFormat_B12
- PixelFormat_R12_Jpeg
- PixelFormat_GR12_Jpeg
- PixelFormat_GB12_Jpeg
- PixelFormat_B12_Jpeg

See also

[spinPixelFormatEnums](#)
[spinImageReset](#)
[spinImageProcessorSetNumDecompressionThreads](#)

Parameters

<i>hImageProcessor</i>	The image processor context
<i>hSrcImageList</i>	List of images from which to convert the images from.
<i>hDestImage</i>	The destination image in which the converted image data will be stored.
<i>destFormat</i>	Output format of the converted image.

13.12.1.114 spinImageProcessorCreate()

```
SPINNAKERC_API spinImageProcessorCreate (
    spinImageProcessor * phImageProcessor )
```

Creates an image processor.

See also

spinError

Parameters

<i>phImageProcessor</i>	The image processor handle pointer in which the image processor context is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.115 spinImageProcessorDestroy()

```
SPINNAKERC_API spinImageProcessorDestroy (
    spinImageProcessor hImageProcessor )
```

Destroys a image list.

See also

spinError

Parameters

<i>hImageProcessor</i>	The image processor context to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.116 spinImageProcessorGetColorProcessing()

```
SPINNAKERC_API spinImageProcessorGetColorProcessing (
    spinImageProcessor hImageProcessor,
    spinColorProcessingAlgorithm * pColorAlgorithm )
```

Gets the default color processing algorithm.

Parameters

<i>hImageProcessor</i>	The image processor context
<i>pColorAlgorithm</i>	The color processing algorithm pointer in which the color processing algorithm is returned

See also

[spinImageProcessorSetColorProcessing\(\)](#)

13.12.1.117 spinImageProcessorGetNumDecompressionThreads()

```
SPINNAKERC_API spinImageProcessorGetNumDecompressionThreads (
    spinImageProcessor hImageProcessor,
    unsigned int * pNumThreads )
```

Gets the number of threads used for image decompression during [spinImageProcessorConvert\(\)](#).

Parameters

<i>hImageProcessor</i>	The image processor context
<i>pNumThreads</i>	The unsigned integer pointer in which the number of parallel image decompression threads is returned

See also

[spinImageProcessorSetNumDecompressionThreads\(\)](#)

13.12.1.118 spinImageProcessorSetColorProcessing()

```
SPINNAKERC_API spinImageProcessorSetColorProcessing (
    spinImageProcessor hImageProcessor,
    spinColorProcessingAlgorithm colorAlgorithm )
```

Sets the color processing algorithm used at the time of the [spinImageProcessorConvert\(\)](#) call, therefore the most recent execution of this function will take precedence.

The DEFAULT algorithm is deprecated and should not be used in the ImageProcessor class.

Parameters

<i>hImageProcessor</i>	The image processor context
<i>colorAlgorithm</i>	The color processing algorithm to set.

See also

[spinImageProcessorGetColorProcessing\(\)](#)

13.12.1.119 spinImageProcessorSetNumDecompressionThreads()

```
SPINNAKERC_API spinImageProcessorSetNumDecompressionThreads (
    spinImageProcessor hImageProcessor,
    unsigned int numThreads )
```

Sets the default number of threads used for image decompression during [spinImageProcessorConvert\(\)](#).

The number of threads used is defaulted to be equal to one less than the number of concurrent threads supported by the system.

Parameters

<i>hImageProcessor</i>	The image processor context
<i>numThreads</i>	Number of parallel image decompression threads set to run

See also

[spinImageProcessorConvert\(\)](#)

13.12.1.120 spinImageRelease()

```
SPINNAKERC_API spinImageRelease (
    spinImage hImage )
```

Releases an image.

See also

[spinError](#)

Parameters

<i>hImage</i>	The image to be saved

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.121 spinImageReset()

```
SPINNAKERC_API spinImageReset (
    spinImage hImage,
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    spinPixelFormatEnums pixelFormat )
```

Resets an image with some set properties.

See also

`spinError`

Parameters

<i>hImage</i>	The image to be reset
<i>width</i>	The width to be reset to
<i>height</i>	The height to be reset to
<i>offsetX</i>	The offset to be reset to along the X axis
<i>offsetY</i>	The offset to be reset to along the Y axis
<i>pixelFormat</i>	The pixel format to be reset to

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.122 spinImageResetEx()

```
SPINNAKERC_API spinImageResetEx (
    spinImage hImage,
    size_t width,
    size_t height,
    size_t offsetX,
```



```

size_t offsetY,
spinPixelFormatEnums pixelFormat,
void * pData )

```

Resets an image with some set properties and image data.

See also

spinError

Parameters

<i>hImage</i>	The image to reset
<i>width</i>	The width to be reset to
<i>height</i>	The height to be reset to
<i>offsetX</i>	The offset to be reset to along the X axis
<i>offsetY</i>	The offset to be reset to along the Y axis
<i>pixelFormat</i>	The pixel format to be reset to
<i>pData</i>	The image data to reset to

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.123 spinImageSave()

```

SPINNAKERC_API spinImageSave (
    spinImage hImage,
    const char * pFilename,
    spinImageFileFormat format )

```

Saves an image using a specified file format (using an enum, spinImageFileFormat)

See also

spinError

spinImageFileFormat

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension) @Param format The file format to use to save the image

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.124 spinImageSaveBmp()

```
SPINNAKERC_API spinImageSaveBmp (
    spinImage hImage,
    const char * pFilename,
    const spinBMPOption * pOption )
```

Saves an image as a BMP image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension)
<i>pOption</i>	The image options related to saving as BMP; includes whether to save as indexed 8-bit

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.125 spinImageSaveFromExt()

```
SPINNAKERC_API spinImageSaveFromExt (
    spinImage hImage,
    const char * pFilename )
```

Saves an image using a specified file format (using the extension of the filename)

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension)

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.126 spinImageSaveJpeg()

```
SPINNAKERC_API spinImageSaveJpeg (
    spinImage hImage,
    const char * pFilename,
    const spinJPEGOption * pOption )
```

Saves an image as a JPEG image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension)
<i>pOption</i>	The image options related to saving as JPEG; includes quality and whether to save as progressive

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.127 spinImageSaveJpg2()

```
SPINNAKERC_API spinImageSaveJpg2 (
    spinImage hImage,
    const char * pFilename,
    const spinJPG2Option * pOption )
```

Saves an image as a JPEG 2000 image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension)
<i>pOption</i>	The image options related to saving as JPEG 2000; includes quality

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.128 spinImageSavePgm()

```
SPINNAKERC_API spinImageSavePgm (
    spinImage hImage,
    const char * pFilename,
    const spinPGMOption * pOption )
```

Saves an image as an PGM image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension)
<i>pOption</i>	The image options related to saving as PGM; includes whether to save as binary

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.129 spinImageSavePng()

```
SPINNAKERC_API spinImageSavePng (
    spinImage hImage,
    const char * pFilename,
    const spinPNGOption * pOption )
```

Saves an image as a PNG image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension)
<i>pOption</i>	The image options related to saving as PNG; includes compression level and whether to save as interlaced

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.130 spinImageSavePpm()

```
SPINNAKERC_API spinImageSavePpm (
    spinImage hImage,
    const char * pFilename,
    const spinPPMOption * pOption )
```

Saves an image as a PPM image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension)
<i>pOption</i>	The image options related to saving as PPM; includes whether to save as binary

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.131 spinImageSaveTiff()

```
SPINNAKERC_API spinImageSaveTiff (
    spinImage hImage,
    const char * pFilename,
    const spinTIFFOption * pOption )
```

Saves an image as a TIFF image.

See also

spinError

Parameters

<i>hImage</i>	The image to be saved
<i>pFilename</i>	The filename to use to save the image (with or without the appropriate file extension)
<i>pOption</i>	The image options related to saving as TIFF; includes compression method

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.132 spinImageStatisticsCreate()

```
SPINNAKERC_API spinImageStatisticsCreate (
    spinImageStatistics * phStatistics )
```

Creates an image statistics context.

Parameters

<i>phStatistics</i>	The statistics handle pointer in which the image statistics context is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.133 spinImageStatisticsDestroy()

```
SPINNAKERC_API spinImageStatisticsDestroy (
    spinImageStatistics hStatistics )
```

Destroys an image statistics context.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.134 spinImageStatisticsDisableAll()

```
SPINNAKERC_API spinImageStatisticsDisableAll (
    spinImageStatistics hStatistics )
```

Disables all channels of an image statistics context.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context to disable all channels

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.135 spinImageStatisticsEnableAll()

```
SPINNAKERC_API spinImageStatisticsEnableAll (  
    spinImageStatistics hStatistics )
```

Enables all channels of an image statistics context.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context to enable all channels

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.136 spinImageStatisticsEnableGreyOnly()

```
SPINNAKERC_API spinImageStatisticsEnableGreyOnly (  
    spinImageStatistics hStatistics )
```

Disables all channels of an image statistics context except grey-scale.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context to enable only grey

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.137 spinImageStatisticsEnableHslOnly()

```
SPINNAKERC_API spinImageStatisticsEnableHslOnly (
    spinImageStatistics hStatistics )
```

Disables all channels of an image statistics context except hue, saturation, and lightness.

See also

`spinError`

Parameters

<i>hStatistics</i>	The image statistics context to enable only HSL

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.138 spinImageStatisticsEnableRgbOnly()

```
SPINNAKERC_API spinImageStatisticsEnableRgbOnly (
    spinImageStatistics hStatistics )
```

Disables all channels of an image statistics context except red, blue, and green.

See also

`spinError`

Parameters

<i>hStatistics</i>	The image statistics context to enable only RGB

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.139 spinImageStatisticsGetAll()

```
SPINNAKERC_API spinImageStatisticsGetAll (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    unsigned int * pRangeMin,
    unsigned int * pRangeMax,
    unsigned int * pPixelValueMin,
    unsigned int * pPixelValueMax,
    unsigned int * pNumPixelValues,
    float * pPixelValueMean,
    int ** ppHistogram )
```

Retrieves all available information of an image statistics channel.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context of the channel
<i>channel</i>	The channel of the information to retrieve
<i>pRangeMin</i>	The unsigned integer pointer in which the minimum value of the range is returned
<i>pRangeMax</i>	The unsigned integer pointer in which the maximum value of the range is returned
<i>pPixelValueMin</i>	The unsigned integer pointer in which the minimum pixel value of the range is returned
<i>pPixelValueMax</i>	The unsigned integer pointer in which the maximum pixel value of the range is returned
<i>pNumPixelValues</i>	The unsigned integer pointer in which the number of pixel values is returned
<i>pPixelValueMean</i>	The float pointer in which the mean pixel value is returned
<i>ppHistogram</i>	The pointer to the pointer in which the histogram data is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.140 spinImageStatisticsGetChannelStatus()

```
SPINNAKERC_API spinImageStatisticsGetChannelStatus (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    bool8_t * pbEnabled )
```

Checks whether an image statistics context is enabled.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context of the channel
<i>channel</i>	The channel to check
<i>pbEnabled</i>	The boolean pointer to return whether or not the channel is enabled

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.141 spinImageStatisticsGetHistogram()

```
SPINNAKERC_API spinImageStatisticsGetHistogram (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    int ** ppHistogram )
```

Retrieves a histogram of an image statistics channel.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context of the channel
<i>channel</i>	The channel of the histogram to be returned
<i>pHistogram</i>	The pointer to the integer pointer in which the histogram data is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.142 spinImageStatisticsGetMean()

```
SPINNAKERC_API spinImageStatisticsGetMean (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    float * pMean )
```

Retrieves the mean of pixel values of an image statistics channel.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context of the channel
<i>channel</i>	The channel of the mean pixel value to be retrieved
<i>pMean</i>	The float pointer in which the mean pixel value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.143 spinImageStatisticsGetNumPixelValues()

```
SPINNAKERC_API spinImageStatisticsGetNumPixelValues (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    unsigned int * pNumValues )
```

Retrieves the number of pixel values of an image statistics channel.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context of the channel
<i>channel</i>	The channel where the pixel values to be counted are
<i>iNumValues</i>	The unsigned integer pointer in which the number of pixel values is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.144 spinImageStatisticsGetPixelValueRange()

```
SPINNAKERC_API spinImageStatisticsGetPixelValueRange (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax )
```

Retrieves the pixel value range of an image statistics channel.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context of the channel
<i>channel</i>	The channel of the pixel value range to retrieve
<i>pMin</i>	The unsigned integer pointer in which the minimum value of the pixel value range is returned
<i>pMax</i>	The unsigned integer pointer in which the maximum value of the pixel value range is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.145 spinImageStatisticsGetRange()

```
SPINNAKERC_API spinImageStatisticsGetRange (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax )
```

Retrieves the range of an image statistics channel.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context of the channel
<i>channel</i>	The channel of the range to retrieve
<i>pMin</i>	The unsigned integer pointer in which the minimum value of the range is returned
<i>pMax</i>	The unsigned integer pointer in which the maximum value of the range is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.146 spinImageStatisticsSetChannelStatus()

```
SPINNAKERC_API spinImageStatisticsSetChannelStatus (
    spinImageStatistics hStatistics,
    spinStatisticsChannel channel,
    bool8_t bEnable )
```

Sets the status of an image statistics channel.

See also

spinError

Parameters

<i>hStatistics</i>	The image statistics context of the channel
<i>channel</i>	The channel to enable/disable
<i>bEnable</i>	The boolean value to set; true enables, false disables

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.147 spinInterfaceEventHandlerCreate()

```
SPINNAKERC_API spinInterfaceEventHandlerCreate (
    spinInterfaceEventHandler * phInterfaceEventHandler,
    spinArrivalEventFunction pArrivalFunction,
    spinRemovalEventFunction pRemovalFunction,
    void * pUserData )
```

Creates an interface event handler (both device arrival and device removal)

See also

spinError

Parameters

<i>phInterfaceEventHandler</i>	The interface event handler pointer in which the interface event context is created
<i>pArrivalFunction</i>	The function to be called at arrival event occurrences; signature to match: void(spinArrivalEventFunction)(void pUserData)
<i>hRemovalFunction</i>	The function to be called at removal event occurrences; signature to match: void(spinRemovalEventFunction)(uint64_t deviceSerialNumber, void pUserData)
<i>pUserData</i>	Properties that can be passed into the event function

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.148 spinInterfaceEventHandlerDestroy()

```
SPINNAKERC_API spinInterfaceEventHandlerDestroy (
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Destroys an interface event handler (both device arrival and device removal)

See also

spinError

Parameters

<i>hInterfaceEventHandler</i>	The interface event handler to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.149 spinInterfaceGetCameras()

```
SPINNAKERC_API spinInterfaceGetCameras (
    spinInterface hInterface,
    spinCameraList hCameraList )
```

Retrieves a camera list from an interface; camera lists must be created and destroy.

See also

[spinCameraListCreateEmpty\(\)](#)

[spinCameraListDestroy\(\)](#)

spinError

Parameters

<i>hInterface</i>	The interface of the camera list to retrieve
<i>hCameraList</i>	The camera list to house the cameras from the interface

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.150 spinInterfaceGetCamerasEx()

```
SPINNAKERC_API spinInterfaceGetCamerasEx (
    spinInterface hInterface,
    bool8_t bUpdateCameras,
    spinCameraList hCameraList )
```

Retrieves a camera list from an interface; manually set whether to update the cameras; camera lists must be created and destroyed.

See also

[spinCameraListCreateEmpty\(\)](#)

[spinCameraListDestroy\(\)](#)

spinError

Parameters

<i>hInterface</i>	The interface of the camera list to retrieve
<i>bUpdateCameras</i>	The boolean of whether or not to update the cameras
<i>hCameraList</i>	The camera list to house the cameras from the interface

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.151 spinInterfaceGetTLNodeMap()

```
SPINNAKERC_API spinInterfaceGetTLNodeMap (
    spinInterface hInterface,
    spinNodeMapHandle * phNodeMap )
```

Retrieves the transport layer nodemap from an interface.

See also

spinError

Parameters

<i>hInterface</i>	The interface of the nodemap to retrieve
<i>phNodeMap</i>	The nodemap handle pointer in which the transport layer interface nodemap is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.152 spinInterfaceIsInUse()

```
SPINNAKERC_API spinInterfaceIsInUse (
    spinInterface hInterface,
    bool8_t * pbIsInUse )
```

Checks whether an interface is in use.

See also

spinError

Parameters

<i>hInterface</i>	The interface to check
<i>pblsInUse</i>	The boolean pointer to return whether or not the interface is in use

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.153 spinInterfaceListClear()

```
SPINNAKERC_API spinInterfaceListClear (
    spinInterfaceList hInterfaceList )
```

Clears an interface list.

See also

`spinError`

Parameters

<i>hInterfaceList</i>	The interface list to clear

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.154 spinInterfaceListCreateEmpty()

```
SPINNAKERC_API spinInterfaceListCreateEmpty (
    spinInterfaceList * phInterfaceList )
```

Creates an empty interface list (interface lists created this way must be destroyed)

See also

`spinError`

Parameters

<i>phInterfaceList</i>	The interface list handle pointer in which the empty interface list is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.155 spinInterfaceListDestroy()

```
SPINNAKERC_API spinInterfaceListDestroy (
    spinInterfaceList hInterfaceList )
```

Destroys an interface list.

See also

spinError

Parameters

<i>hInterfaceList</i>	The interface list to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.156 spinInterfaceListGet()

```
SPINNAKERC_API spinInterfaceListGet (
    spinInterfaceList hInterfaceList,
    size_t index,
    spinInterface * phInterface )
```

Retrieves an interface from an interface list using an index (interfaces retrieved this way must be released)

See also

spinError

Parameters

<i>hInterfaceList</i>	The interface list of the interface to be retrieved
<i>index</i>	The index of the interface
<i>phInterface</i>	The interface handle pointer in which the interface is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.157 spinInterfaceListGetSize()

```
SPINNAKERC_API spinInterfaceListGetSize (
    spinInterfaceList hInterfaceList,
    size_t * pSize )
```

Retrieves the number of interfaces in an interface list.

See also

spinError

Parameters

<i>hInterfaceList</i>	The interface list where the interfaces to be counted are
<i>pSize</i>	The unsigned integer pointer in which the number of interfaces is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

See also

spinError

13.12.1.158 spinInterfaceRegisterDeviceArrivalEventHandler()

```
SPINNAKERC_API spinInterfaceRegisterDeviceArrivalEventHandler (
    spinInterface hInterface,
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Registers a device arrival event handler on an interface (event handlers registered in this way must be unregistered)

See also

spinError

Parameters

<i>hInterface</i>	The interface on which to register the device arrival event handler
<i>hDeviceArrivalEventHandler</i>	The device arrival event handler to register

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.159 spinInterfaceRegisterDeviceRemovalEventHandler()

```
SPINNAKERC_API spinInterfaceRegisterDeviceRemovalEventHandler (
    spinInterface hInterface,
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Registers a device removal event handler on an interface (event handlers registered in this way must be unregistered)

See also

spinError

Parameters

<i>hInterface</i>	the Interface on which to register the device removal event handler
<i>hDeviceRemovalEventHandler</i>	The device removal event handler to register

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.160 spinInterfaceRegisterInterfaceEventHandler()

```
SPINNAKERC_API spinInterfaceRegisterInterfaceEventHandler (
    spinInterface hInterface,
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Registers an interface event handler (both device arrival and device removal) on an interface.

See also

spinError

Parameters

<i>hInterface</i>	The interface on which to register the interface event handler
<i>hInterfaceEventHandler</i>	The interface event handler to register

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.161 spinInterfaceRelease()

```
SPINNAKERC_API spinInterfaceRelease (
    spinInterface hInterface )
```

Releases an interface.

See also

`spinError`

Parameters

<i>hInterface</i>	The interface to release

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.162 spinInterfaceSendActionCommand()

```
SPINNAKERC_API spinInterfaceSendActionCommand (
    spinInterface hInterface,
    size_t iDeviceKey,
    size_t iGroupKey,
    size_t iGroupMask,
    size_t iActionTime,
    size_t * piResultSize,
    actionCommandResult results[] )
```

Broadcast an Action Command to all devices on interface.

See also

`spinError`

Parameters

<i>iDeviceKey</i>	The Action Command's device key
<i>iGroupKey</i>	The Action Command's group key
<i>iGroupMask</i>	The Action Command's group mask
<i>iActionTime</i>	(Optional) Time when to assert a future action. Zero means immediate action.

Parameters

<i>piResultSize</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 *piResultSize 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 piResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.163 spinInterfaceUnregisterDeviceArrivalEventHandler()

```
SPINNAKERC_API spinInterfaceUnregisterDeviceArrivalEventHandler (
    spinInterface hInterface,
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Unregisters a device arrival event handler from an interface.

See also

spinError

Parameters

<i>hInterface</i>	The interface from which to unregister the device arrival event handler
<i>hDeviceArrivalEventHandler</i>	The device arrival event handler to unregister

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.164 spinInterfaceUnregisterDeviceRemovalEventHandler()

```
SPINNAKERC_API spinInterfaceUnregisterDeviceRemovalEventHandler (
    spinInterface hInterface,
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Unregisters a device removal event handler from an interface.

See also

spinError

Parameters

<i>hInterface</i>	The interface from which to unregister the device removal event handler
<i>hDeviceRemovalEventHandler</i>	The device removal event handler to unregister

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.165 spinInterfaceUnregisterInterfaceEventHandler()

```
SPINNAKERC_API spinInterfaceUnregisterInterfaceEventHandler (  
    spinInterface hInterface,  
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Unregisters an interface event handler from an interface.

See also

spinError

Parameters

<i>hInterface</i>	The interface from which to unregister the interface event handler
<i>hInterfaceEventHandler</i>	The interface event handler to unregister

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.166 spinInterfaceUpdateCameras()

```
SPINNAKERC_API spinInterfaceUpdateCameras (  
    spinInterface hInterface,  
    bool8_t * pbChanged )
```

Checks whether any cameras have been connected or disconnected on an interface.

See also

spinError

Parameters

<i>hInterface</i>	The interface of the list of attached cameras to update
<i>pbChanged</i>	The boolean pointer to return whether or not the cameras have changed

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.167 spinLogDataGetCategoryName()

```
SPINNAKERC_API spinLogDataGetCategoryName (
    spinLogEventData hLogEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the category name of a log event.

See also

spinError

Parameters

<i>hLogEventData</i>	The log event data received from the log event
<i>pBuf</i>	The c-string character buffer in which the category name of the log event is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.168 spinLogDataGetLogMessage()

```
SPINNAKERC_API spinLogDataGetLogMessage (
    spinLogEventData hLogEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the log message of a log event.

See also

spinError

Parameters

<i>hLogEventData</i>	The log event data received from the log event
<i>pBuf</i>	The c-string character buffer in which the log message of the log event is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.169 spinLogDataGetNDC()

```
SPINNAKERC_API spinLogDataGetNDC (
    spinLogEventData hLogEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the NDC of a log event.

See also

`spinError`

Parameters

<i>hLogEventData</i>	The log event data received from the log event
<i>pBuf</i>	The c-string character buffer in which the NDC of the log event is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.170 spinLogDataGetPriority()

```
SPINNAKERC_API spinLogDataGetPriority (
    spinLogEventData hLogEventData,
    int64_t * pValue )
```

Retrieves the priority of a log event.

See also

`spinError`

Parameters

<i>hLogEventData</i>	The log event data received from the log event
<i>pValue</i>	The integer pointer in which the priority value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.171 spinLogDataGetPriorityName()

```
SPINNAKERC_API spinLogDataGetPriorityName (  
    spinLogEventData hLogEventData,  
    char * pBuf,  
    size_t * pBufLen )
```

Retrieves the priority name of a log event.

See also

spinError

Parameters

<i>hLogEventData</i>	The log event data received from the log event
<i>pBuf</i>	The c-string character buffer in which the priority name of the log event is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.172 spinLogDataGetThreadName()

```
SPINNAKERC_API spinLogDataGetThreadName (  
    spinLogEventData hLogEventData,  
    char * pBuf,  
    size_t * pBufLen )
```

Retrieves the thread name of a log event.

See also

spinError

Parameters

<i>hLogEventData</i>	The log event data received from the log event
<i>pBuf</i>	The c-string character buffer in which the thread name of the log event is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.173 spinLogDataGetTimestamp()

```
SPINNAKERC_API spinLogDataGetTimestamp (
    spinLogEventData hLogEventData,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the timestamp of a log event.

See also

`spinError`

Parameters

<i>hLogEventData</i>	The log event data received from the log event
<i>pBuf</i>	The c-string character buffer in which the timestamp of the log event is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.174 spinLogEventHandlerCreate()

```
SPINNAKERC_API spinLogEventHandlerCreate (
    spinLogEventHandler * phLogEventHandler,
    spinLogEventFunction pFunction,
    void * pUserData )
```

Creates a log event handler.

See also

`spinError`

Parameters

<i>phLogEventHandler</i>	The log event handler pointer in which the log event context is created
<i>pFunction</i>	The function to be called at device event occurrences; signature to match: void(spinLogEventFunction)(const spinLogEventData hEventData, void pUserData)
<i>pUserData</i>	Properties that can be passed into the event function

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.175 spinLogEventHandlerDestroy()

```
SPINNAKERC_API spinLogEventHandlerDestroy (  
    spinLogEventHandler hLogEventHandler )
```

Destroys a log event handler.

See also

spinError

Parameters

<i>hLogEventHandler</i>	The log event handler to destroy

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.176 spinSystemGetCameras()

```
SPINNAKERC_API spinSystemGetCameras (  
    spinSystem hSystem,  
    spinCameraList hCameraList )
```

Retrieves a list of detected (and enumerable) cameras on the system; camera lists must be created and destroyed.

See also

[spinCameraListCreateEmpty\(\)](#)

[spinCameraListDestroy\(\)](#)

spinError

Parameters

<i>hSystem</i>	The system, from which the camera list is retrieved
<i>hCameraList</i>	The camera list to house the cameras from the system

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.177 spinSystemGetCamerasEx()

```
SPINNAKERC_API spinSystemGetCamerasEx (
    spinSystem hSystem,
    bool8_t bUpdateInterfaces,
    bool8_t bUpdateCameras,
    spinCameraList hCameraList )
```

Retrieves a list of detected (and enumerable) cameras on the system; manually set whether to update the current interface and camera lists; camera lists must be created and destroyed.

See also

[spinCameraListCreateEmpty\(\)](#)

[spinCameraListDestroy\(\)](#)

spinError

Parameters

<i>hSystem</i>	The system, from which the camera list is retrieved
<i>bUpdateInterfaces</i>	The boolean of whether to update the interface list
<i>bUpdateCameras</i>	The boolean of whether to update the camera list
<i>hCameraList</i>	The camera list to house the cameras from the system

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.178 spinSystemGetInstance()

```
SPINNAKERC_API spinSystemGetInstance (
    spinSystem * phSystem )
```

Retrieves an instance of the system object; the system is a singleton, so there will only ever be one instance; system instance must be destroyed by calling `spinSystemReleaseInstance`.

See also

[spinSystemReleaseInstance](#)
[spinError](#)

Parameters

<i>phSystem</i>	The system handle pointer in which the system instance is returned

Returns

[spinError](#) The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.179 spinSystemGetInterfaces()

```
SPINNAKERC_API spinSystemGetInterfaces (
    spinSystem hSystem,
    spinInterfaceList hInterfaceList )
```

Retrieves a list of detected (and enumerable) interfaces on the system; interface lists must be created and destroyed.

See also

[spinInterfaceListCreateEmpty\(\)](#)
[spinInterfaceListDestroy\(\)](#)
[spinError](#)

Parameters

<i>hSystem</i>	The system, from which the interface list is retrieved
<i>hInterfaceList</i>	The interface list to house the interfaces from the system

Returns

[spinError](#) The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.180 spinSystemGetLibraryVersion()

```
SPINNAKERC_API spinSystemGetLibraryVersion (
    spinSystem hSystem,
    spinLibraryVersion * hLibraryVersion )
```

Get current library version of Spinnaker.

Returns

A struct containing the current version of Spinnaker(major, minor, type, build).

13.12.1.181 spinSystemGetLoggingLevel()

```
SPINNAKERC_API spinSystemGetLoggingLevel (
    spinSystem hSystem,
    spinnakerLogLevel * pLogLevel )
```

Retrieves the logging level for all logging events on the system.

See also

spinError

Parameters

<i>hSystem</i>	The system, from which the logging level is retrieved
<i>logLevel</i>	The logging level enum pointer in which the current logging level is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.182 spinSystemGetTLNodeMap()

```
SPINNAKERC_API spinSystemGetTLNodeMap (
    spinSystem hSystem,
    spinNodeMapHandle * phNodeMap )
```

Retrieves the transport layer nodemap from the system.

See also

spinError

Parameters

<i>hSystem</i>	The system handle.
<i>phNodeMap</i>	The nodemap handle pointer in which the transport layer system nodemap is returned.

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.183 spinSystemIsInUse()

```
SPINNAKERC_API spinSystemIsInUse (
    spinSystem hSystem,
    bool8_t * pbIsInUse )
```

Checks whether a system is currently in use.

See also

spinError

Parameters

<i>hSystem</i>	The system to check
<i>pbIsInUse</i>	The boolean pointer to return whether the system is currently in use

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.184 spinSystemRegisterDeviceArrivalEventHandler()

```
SPINNAKERC_API spinSystemRegisterDeviceArrivalEventHandler (
    spinSystem hSystem,
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Registers a device arrival event handler to every interface on the system (event handlers registered this way must be unregistered)

See also

spinError

Parameters

<i>hSystem</i>	The system, on which the device arrival event handler is registered
<i>hDeviceArrivalEventHandler</i>	The device arrival event handler to register on the system

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.185 spinSystemRegisterDeviceRemovalEventHandler()

```
SPINNAKERC_API spinSystemRegisterDeviceRemovalEventHandler (
    spinSystem hSystem,
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Registers a device removal event handler to the system to every interface on the system (event handlers registered this way must be unregistered)

See also

spinError

Parameters

<i>hSystem</i>	The system, on which the device removal event handler is registered
<i>hDeviceRemovalEventHandler</i>	The device removal event handler to register on the system

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.186 spinSystemRegisterInterfaceEventHandler()

```
SPINNAKERC_API spinSystemRegisterInterfaceEventHandler (
    spinSystem hSystem,
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Registers an interface event handler (device arrival and device removal) to every interface on the system (interface events registered this way must be unregistered) If new interfaces are detected by the system after [spinSystemRegisterInterfaceEventHandler\(\)](#) is called, those interfaces will be automatically registered with this event.

See also

spinError

[spinInterfaceEventHandler](#)

Parameters

<i>hSystem</i>	The system, on which the interface event handler is registered
<i>hInterfaceEventHandler</i>	The interface event handler (device arrival and device removal) to register on the system

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.187 spinSystemRegisterLogEventHandler()

```
SPINNAKERC_API spinSystemRegisterLogEventHandler (
    spinSystem hSystem,
    spinLogEventHandler hLogEventHandler )
```

Registers a logging event handler to the system (event handlers registered in this way must be unregistered)

See also

[spinError](#)

Parameters

<i>hSystem</i>	The system, on which the logging event handler is registered
<i>hLogEventHandler</i>	The logging event handler to register on the system

Returns

[spinError](#) The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.188 spinSystemReleaseInstance()

```
SPINNAKERC_API spinSystemReleaseInstance (
    spinSystem hSystem )
```

Releases the system; make sure handle is cleaned up properly by setting it to NULL after system is released; the handle can only be used again after calling [spinSystemGetInstance](#).

See also

[spinSystemGetInstance](#)

[spinError](#)

Parameters

<i>hSystem</i>	The system handle

Returns

[spinError](#) The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.189 spinSystemSendActionCommand()

```
SPINNAKERC_API spinSystemSendActionCommand (
    spinSystem hSystem,
```

```

size_t iDeviceKey,
size_t iGroupKey,
size_t iGroupMask,
size_t iActionTime,
size_t * piResultSize,
actionCommandResult results[] )

```

Broadcast an Action Command to all devices on system.

See also

spinError

Parameters

<i>hSystem</i>	The system on which to send the action command to all devices.
<i>iDeviceKey</i>	The Action Command's device key
<i>iGroupKey</i>	The Action Command's group key
<i>iGroupMask</i>	The Action Command's group mask
<i>iActionTime</i>	(Optional) Time when to assert a future action. Zero means immediate action.
<i>piResultSize</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 *piResultSize 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 piResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.190 spinSystemSetLoggingLevel()

```

SPINNAKERC_API spinSystemSetLoggingLevel (
    spinSystem hSystem,
    spinnakerLogLevel logLevel )

```

Sets the logging level for all logging events on the system.

See also

spinError

Parameters

<i>hSystem</i>	The system, on which the logging level is set
<i>logLevel</i>	The logging level to set

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.191 spinSystemUnregisterAllLogEventHandlers()

```
SPINNAKERC_API spinSystemUnregisterAllLogEventHandlers (  
    spinSystem hSystem )
```

Unregisters all logging event handlers from the system.

See also

spinError

Parameters

<i>hSystem</i>	The system, from which all logging event handlers are unregistered

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.192 spinSystemUnregisterDeviceArrivalEventHandler()

```
SPINNAKERC_API spinSystemUnregisterDeviceArrivalEventHandler (  
    spinSystem hSystem,  
    spinDeviceArrivalEventHandler hDeviceArrivalEventHandler )
```

Unregisters a device arrival event handler from the system.

See also

spinError

[spinDeviceArrivalEventHandler](#)

Parameters

<i>hSystem</i>	The system, from which the device arrival event handler is unregistered
<i>hDeviceArrivalEventHandler</i>	The device arrival event handler to unregister from the system

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.193 spinSystemUnregisterDeviceRemovalEventHandler()

```
SPINNAKERC_API spinSystemUnregisterDeviceRemovalEventHandler (
    spinSystem hSystem,
    spinDeviceRemovalEventHandler hDeviceRemovalEventHandler )
```

Unregisters a device removal event handler from the system.

See also

[spinError](#)

[spinDeviceRemovalEventHandler](#)

Parameters

<i>hSystem</i>	The system, from which the device removal event handler is unregistered
<i>hDeviceRemovalEventHandler</i>	The device removal event handler to unregister from the system

Returns

[spinError](#) The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.194 spinSystemUnregisterInterfaceEventHandler()

```
SPINNAKERC_API spinSystemUnregisterInterfaceEventHandler (
    spinSystem hSystem,
    spinInterfaceEventHandler hInterfaceEventHandler )
```

Unregisters an interface event handler from the system.

See also

[spinError](#)

[spinInterfaceEventHandler](#)

Parameters

<i>hSystem</i>	The system, from which the interface event handler is unregistered
<i>hInterfaceEventHandler</i>	The interface event handler (device arrival and device removal) to unregister from the system

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.195 spinSystemUnregisterLogEventHandler()

```
SPINNAKERC_API spinSystemUnregisterLogEventHandler (
    spinSystem hSystem,
    spinLogEventHandler hLogEventHandler )
```

Unregisters a selected logging event handler from the system.

See also

spinError

Parameters

<i>hSystem</i>	The system, from which the logging event handler is unregistered
<i>hLogEventHandler</i>	The logging event handler to unregister from the system

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.12.1.196 spinSystemUpdateCameras()

```
SPINNAKERC_API spinSystemUpdateCameras (
    spinSystem hSystem,
    bool8_t * pbChanged )
```

Updates the list of cameras on the system, informing whether there has been any changes.

See also

spinError

Parameters

<i>hSystem</i>	The system, on which the list of attached cameras is updated
<i>pbChanged</i>	The boolean pointer to return whether cameras have arrived on or been removed from the system

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.12.1.197 spinSystemUpdateCamerasEx()

```
SPINNAKERC_API spinSystemUpdateCamerasEx (
    spinSystem hSystem,
    bool8_t bUpdateInterfaces,
    bool8_t * pbChanged )
```

Updates the list of cameras on the system, informing whether there has been any changes; manually set whether to update the current interface lists.

See also

`spinError`

Parameters

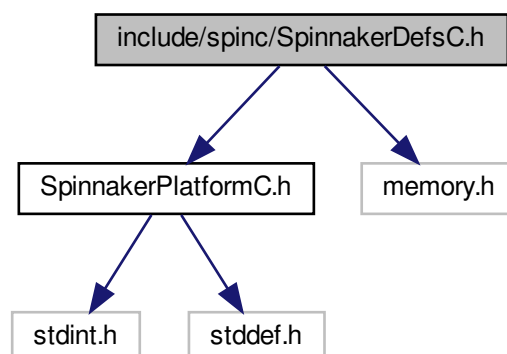
<i>hSystem</i>	The system, on which the list of attached cameras is updated
<i>bUpdateInterfaces</i>	The boolean of whether to update the interface list
<i>pbChanged</i>	The boolean pointer to return whether cameras have arrived or been removed from the system

Returns

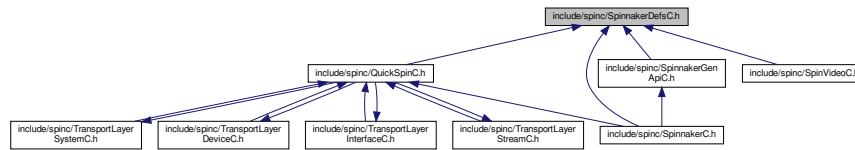
`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.13 include/spinc/SpinnakerDefsC.h File Reference

Include dependency graph for `SpinnakerDefsC.h`:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [_spinPNGOption](#)
Options for saving PNG images.
- struct [_spinPPMOption](#)
Options for saving PPM images.
- struct [_spinPGMOption](#)
Options for saving PGM images.
- struct [_spinTIFFOption](#)
Options for saving TIFF images.
- struct [_spinJPEGOption](#)
Options for saving JPEG images.
- struct [_spinJPG2Option](#)
Options for saving JPEG 2000 images.
- struct [_spinBMPOption](#)
Options for saving BMP images.
- struct [_spinMJPEGOption](#)
Options for saving MJPG videos.
- struct [_spinH264Option](#)
Options for saving H264 videos.
- struct [_spinAVIOption](#)
Options for saving uncompressed videos.
- struct [_spinLibraryVersion](#)
Provides easier access to the current version of Spinnaker.
- struct [_actionCommandResult](#)
Action Command Result.

Typedefs

- typedef uint8_t [bool8_t](#)
- typedef void * [spinSystem](#)
Handle for system functionality.
- typedef void * [spinInterfaceList](#)
Handle for interface list functionality.
- typedef void * [spinInterface](#)
Handle for interface functionality.
- typedef void * [spinCameraList](#)
Handle for interface functionality.
- typedef void * [spinCamera](#)
Handle for camera functionality.

- typedef void * [spinImage](#)
Handle for image functionality.
- typedef void * [spinImageList](#)
Handle for image list functionality.
- typedef void * [spinImageProcessor](#)
Handle for image processor functionality.
- typedef void * [spinImageStatistics](#)
Handle for image statistics functionality.
- typedef void * [spinDeviceEventHandler](#)
Handle for device event handler functionality.
- typedef void * [spinImageEventHandler](#)
Handle for image event handler functionality.
- typedef void * [spinImageListEventHandler](#)
Handle for image list event handler functionality.
- typedef void * [spinDeviceArrivalEventHandler](#)
Handle for arrival event handler functionality.
- typedef void * [spinDeviceRemovalEventHandler](#)
Handle for removal event handler functionality.
- typedef void * [spinInterfaceEventHandler](#)
Handle for interface event handler functionality.
- typedef void * [spinLogEventHandler](#)
Handle for logging event handler functionality.
- typedef void * [spinLogEventData](#)
Handle for logging event data functionality.
- typedef void * [spinDeviceEventData](#)
Handle for device event data functionality.
- typedef void * [spinVideo](#)
Handle for video recording functionality.
- typedef void(* [spinDeviceEventFunction](#)) (const [spinDeviceEventData](#) hEventData, const char *pEventName, void *pUserData)
Function signatures are used to create and trigger callbacks and events.
- typedef void(* [spinImageEventFunction](#)) (const [spinImage](#) hImage, void *pUserData)
- typedef void(* [spinImageListEventFunction](#)) (const [spinImageList](#) hImage, void *pUserData)
- typedef void(* [spinArrivalEventFunction](#)) (const [spinCamera](#) hCamera, void *pUserData)
- typedef void(* [spinRemovalEventFunction](#)) (const [spinCamera](#) hCamera, void *pUserData)
- typedef void(* [spinLogEventFunction](#)) (const [spinLogEventData](#) hEventData, void *pUserData)

Enumerations

- enum [_spinError](#) {
[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,
SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT = -2001,
SPINNAKER_ERR_GENICAM_OUT_OF_RANGE = -2002,
SPINNAKER_ERR_GENICAM_PROPERTY = -2003,
SPINNAKER_ERR_GENICAM_RUN_TIME = -2004,
SPINNAKER_ERR_GENICAM_LOGICAL = -2005,
SPINNAKER_ERR_GENICAM_ACCESS = -2006,
SPINNAKER_ERR_GENICAM_TIMEOUT = -2007,
SPINNAKER_ERR_GENICAM_DYNAMIC_CAST = -2008,
SPINNAKER_ERR_GENICAM_GENERIC = -2009,
SPINNAKER_ERR_GENICAM_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_CUSTOM_ID = -10000 }

```

The error codes used in Spinnaker C.

- enum `_spinColorProcessingAlgorithm` {


```

SPINNAKER_COLOR_PROCESSING_ALGORITHM_NONE,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR_AVG,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILINEAR,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSING,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQ_LINEAR,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPP,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECTIONAL_FILTER,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGOROUS,
SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEIGHTED_DIRECTIONAL_FILTER }

```

Color processing algorithms.

- enum `_spinStatisticsChannel` {


```

SPINNAKER_STATISTICS_CHANNEL_GREY,
SPINNAKER_STATISTICS_CHANNEL_RED,
SPINNAKER_STATISTICS_CHANNEL_GREEN,
SPINNAKER_STATISTICS_CHANNEL_BLUE,
SPINNAKER_STATISTICS_CHANNEL_HUE,
SPINNAKER_STATISTICS_CHANNEL_SATURATION,
SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS,
SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS }

```

Channels that allow statistics to be calculated.

- enum `_spinImageFileFormat` {


```

SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT = -1,
SPINNAKER_IMAGE_FILE_FORMAT_PGM,
SPINNAKER_IMAGE_FILE_FORMAT_PPM,
SPINNAKER_IMAGE_FILE_FORMAT_BMP,

```

```
SPINNAKER_IMAGE_FILE_FORMAT_JPEG,
SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000,
SPINNAKER_IMAGE_FILE_FORMAT_TIFF,
SPINNAKER_IMAGE_FILE_FORMAT_PNG,
SPINNAKER_IMAGE_FILE_FORMAT_RAW,
SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS = 0x7FFFFFFF }
```

File formats to be used for saving images to disk.

- enum `_spinTLPixelFormatNamespace` {


```
SPINNAKER_TLPixelFormatNamespace_UNKNOWN = 0,
SPINNAKER_TLPixelFormatNamespace_GEV = 1,
SPINNAKER_TLPixelFormatNamespace_IIDC = 2,
SPINNAKER_TLPixelFormatNamespace_PFNC_16BIT = 3,
SPINNAKER_TLPixelFormatNamespace_PFNC_32BIT = 4,
SPINNAKER_PixelFormatNamespace_CUSTOM_ID = 1000 }
```

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

- enum `_spinImageStatus` {


```
SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR = -1,
SPINNAKER_IMAGE_STATUS_NO_ERROR = 0,
SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED = 1,
SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW = 2,
SPINNAKER_IMAGE_STATUS_MISSING_PACKETS,
SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT,
SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT,
SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT,
SPINNAKER_IMAGE_STATUS_MISSING_LEADER = 7,
SPINNAKER_IMAGE_STATUS_MISSING_TRAILER = 8,
SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE = 9,
SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT = 10,
SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID = 11,
SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES = 12 }
```

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

- enum `_spinLogLevel` {


```
SPINNAKER_LOG_LEVEL_OFF = -1,
SPINNAKER_LOG_LEVEL_FATAL = 0,
SPINNAKER_LOG_LEVEL_ALERT = 100,
SPINNAKER_LOG_LEVEL_CRIT = 200,
SPINNAKER_LOG_LEVEL_ERROR = 300,
SPINNAKER_LOG_LEVEL_WARN = 400,
SPINNAKER_LOG_LEVEL_NOTICE = 500,
SPINNAKER_LOG_LEVEL_INFO = 600,
SPINNAKER_LOG_LEVEL_DEBUG = 700,
SPINNAKER_LOG_LEVEL_NOTSET = 800 }
```

log levels

- enum `_spinTLPayloadType` {


```
SPINNAKER_TLPayloadType_UNKNOWN = 0,
SPINNAKER_TLPayloadType_IMAGE = 1,
SPINNAKER_TLPayloadType_RAW_DATA = 2,
SPINNAKER_TLPayloadType_FILE = 3,
SPINNAKER_TLPayloadType_CHUNK_DATA = 4,
SPINNAKER_TLPayloadType_JPEG = 5,
SPINNAKER_TLPayloadType_JPEG2000 = 6,
SPINNAKER_TLPayloadType_H264 = 7,
SPINNAKER_TLPayloadType_CHUNK_ONLY = 8,
SPINNAKER_TLPayloadType_DEVICE_SPECIFIC = 9,
SPINNAKER_TLPayloadType_MULTI_PART = 10,
SPINNAKER_TLPayloadType_CUSTOM_ID = 1000,
SPINNAKER_TLPayloadType_LOSSLESS_COMPRESSED = SPINNAKER_TLPayloadType_CUSTOM_ID + 1,
```

```
SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED = SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID + 2,
SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED = SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID + 3 }
```

- enum `_spinTIFFCompressionMethod` {
`SPINNAKER_TIFF_COMPRESS_METHOD_NONE` = 1,
`SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS`,
`SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE`,
`SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE`,
`SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3`,
`SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4`,
`SPINNAKER_TIFF_COMPRESS_METHOD_LZW`,
`SPINNAKER_TIFF_COMPRESS_METHOD_JPG` }
Compression method to use for encoding TIFF images.
- enum `_spinActionCommandStatus` {
`SPINNAKER_ACTION_COMMAND_STATUS_OK` = 0,
`SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME` = 0x8013,
`SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,
`SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE` = 0x8016,
`SPINNAKER_ACTION_COMMAND_STATUS_ERROR` = 0x8FFF }
Possible Status Codes Returned from Action Command.

Variables

- static const `bool8_t False` = 0
- static const `bool8_t True` = 1

13.13.1 Typedef Documentation

13.13.1.1 `bool8_t`

```
typedef uint8_t bool8_t
```

13.13.1.2 `spinArrivalEventFunction`

```
typedef void(* spinArrivalEventFunction) (const spinCamera hCamera, void *pUserData)
```

13.13.1.3 `spinCamera`

```
typedef void* spinCamera
```

Handle for camera functionality.

Created by calling `spinCameraListGet()`, which requires a call to `spinCameraRelease()` to release.

13.13.1.4 spinCameraList

```
typedef void* spinCameraList
```

Handle for interface functionality.

Created by calling [spinSystemGetCameras\(\)](#) or [spinInterfaceGetCameras\(\)](#), which require a call to [spinCameraListClear\(\)](#) to clear, or [spinCameraListCreateEmpty\(\)](#), which requires a call to [spinCameraListDestroy\(\)](#) to destroy.

13.13.1.5 spinDeviceArrivalEventHandler

```
typedef void* spinDeviceArrivalEventHandler
```

Handle for arrival event handler functionality.

Created by calling [spinArrivalEventCreate\(\)](#), which requires a call to [spinDeviceArrivalEventHandlerDestroy\(\)](#) to destroy.

13.13.1.6 spinDeviceEventData

```
typedef void* spinDeviceEventData
```

Handle for device event data functionality.

Received in device event function. No need to release, clear, or destroy.

13.13.1.7 spinDeviceEventFunction

```
typedef void(* spinDeviceEventFunction) (const spinDeviceEventData hEventData, const char *p↔  
EventName, void *pUserData)
```

Function signatures are used to create and trigger callbacks and events.

13.13.1.8 spinDeviceEventHandler

```
typedef void* spinDeviceEventHandler
```

Handle for device event handler functionality.

Created by calling [spinDeviceEventHandlerCreate\(\)](#), which requires a call to [spinDeviceEventHandlerDestroy\(\)](#) to destroy.

13.13.1.9 spinDeviceRemovalEventHandler

```
typedef void* spinDeviceRemovalEventHandler
```

Handle for removal event handler functionality.

Created by calling [spinDeviceRemovalEventHandlerCreate\(\)](#), which requires a call to [spinDeviceRemovalEventHandlerDestroy\(\)](#) to destroy.

13.13.1.10 spinImage

```
typedef void* spinImage
```

Handle for image functionality.

Created by calling [spinCameraGetNextImage\(\)](#) or [spinCameraGetNextImageEx\(\)](#), which require a call to [spinImageRelease\(\)](#) to remove from buffer, or [spinImageCreateEmpty\(\)](#), [spinImageCreateEx\(\)](#), or [spinImageCreate\(\)](#), which require a call to [spinImageDestroy\(\)](#) to destroy.

13.13.1.11 spinImageEventFunction

```
typedef void(* spinImageEventFunction) (const spinImage hImage, void *pUserData)
```

13.13.1.12 spinImageEventHandler

```
typedef void* spinImageEventHandler
```

Handle for image event handler functionality.

Created by calling [spinImageEventHandlerCreate\(\)](#), which requires a call to [spinImageEventHandlerDestroy\(\)](#) to destroy.

13.13.1.13 spinImageList

```
typedef void* spinImageList
```

Handle for image list functionality.

Created by calling [spinCameraGetNextImageSync\(\)](#), which require a call to [spinImageRelease\(\)](#) to remove from buffer, or [spinImageCreateEmpty\(\)](#), [spinImageCreateEx\(\)](#), or [spinImageCreate\(\)](#), which require a call to [spinImageDestroy\(\)](#) to destroy.

13.13.1.14 spinImageListEventFunction

```
typedef void(* spinImageListEventFunction) (const spinImageList hImage, void *pUserData)
```

13.13.1.15 spinImageListEventHandler

```
typedef void* spinImageListEventHandler
```

Handle for image list event handler functionality.

Created by calling [spinImageListEventHandlerCreate\(\)](#), which requires a call to [spinImageListEventHandlerDestroy\(\)](#) to destroy.

13.13.1.16 spinImageProcessor

```
typedef void* spinImageProcessor
```

Handle for image processor functionality.

Created by calling [spinImageProcessorCreate\(\)](#), which requires a call to [spinImageProcessorDestroy\(\)](#) to destroy.

13.13.1.17 spinImageStatistics

```
typedef void* spinImageStatistics
```

Handle for image statistics functionality.

Created by calling [spinImageStatisticsCreate\(\)](#), which requires a call to [spinImageStatisticsDestroy\(\)](#) to destroy.

13.13.1.18 spinInterface

```
typedef void* spinInterface
```

Handle for interface functionality.

Created by calling [spinInterfaceListGet\(\)](#), which requires a call to [spinInterfaceRelease\(\)](#) to release.

13.13.1.19 spinInterfaceEventHandler

```
typedef void* spinInterfaceEventHandler
```

Handle for interface event handler functionality.

Created by calling [spinInterfaceEventHandlerCreate\(\)](#), which requires a call to [spinInterfaceEventHandlerDestroy\(\)](#) to destroy.

13.13.1.20 spinInterfaceList

```
typedef void* spinInterfaceList
```

Handle for interface list functionality.

Created by calling [spinSystemGetInterfaces\(\)](#), which requires a call to [spinInterfaceListClear\(\)](#) to clear, or [spinInterfaceListCreateEmpty\(\)](#), which requires a call to [spinInterfaceListDestroy\(\)](#) to destroy.

13.13.1.21 spinLogEventData

```
typedef void* spinLogEventData
```

Handle for logging event data functionality.

Received in log event function. No need to release, clear, or destroy.

13.13.1.22 spinLogEventFunction

```
typedef void(* spinLogEventFunction) (const spinLogEventData hEventData, void *pUserData)
```

13.13.1.23 spinLogEventHandler

```
typedef void* spinLogEventHandler
```

Handle for logging event handler functionality.

Created by calling [spinLogEventHandlerCreate\(\)](#), which requires a call to [spinLogEventHandlerDestroy\(\)](#) to destroy.

13.13.1.24 spinRemovalEventFunction

```
typedef void(* spinRemovalEventFunction) (const spinCamera hCamera, void *pUserData)
```

13.13.1.25 spinSystem

```
typedef void* spinSystem
```

Handle for system functionality.

Created by calling [spinSystemGetInstance\(\)](#), which requires a call to [spinSystemReleaseInstance\(\)](#) to release.

13.13.1.26 spinVideo

```
typedef void* spinVideo
```

Handle for video recording functionality.

Created by calling [spinVideoOpenUncompressed\(\)](#), [spinVideoOpenMJPEG\(\)](#), and [spinVideoOpenH264\(\)](#), which require a call to [spinVideoClose\(\)](#) to destroy.

13.13.2 Enumeration Type Documentation

13.13.2.1 _spinActionCommandStatus

```
enum _spinActionCommandStatus
```

Possible Status Codes Returned from Action Command.

Enumerator

SPINNAKER_ACTION_COMMAND_STATUS_OK	The device acknowledged the command.
SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME	
SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW	
SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE	
SPINNAKER_ACTION_COMMAND_STATUS_ERROR	

13.13.2.2 `_spinColorProcessingAlgorithm`

enum `_spinColorProcessingAlgorithm`

Color processing algorithms.

Please refer to our knowledge base at article at <https://www.flir.com/support-center/iis/machine-vision/k> for complete details for each algorithm.

Enumerator

SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_NONE	No color processing.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_NEAREST_NEIGHBOR	Fastest but lowest quality. Equivalent to FLYCAPTURE_NEAREST_NEIGHBOR_FAST in FlyCapture.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_NEAREST_NEIGHBOR_AVG	Nearest Neighbor with averaged green pixels. Higher quality but slower compared to nearest neighbor without averaging.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_BILINEAR	Weighted average of surrounding 4 pixels in a 2x2 neighborhood.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_EDGE_SENSING	Weights surrounding pixels based on localized edge orientation.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_HQ_LINEAR	Well-balanced speed and quality.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_IPP	Multi-threaded with similar results to edge sensing.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_DIRECTIONAL_FILTER	Best quality but much faster than rigorous.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_RIGOROUS	Slowest but produces good results.
SPINNAKER_COLOR_PROCESSING_ALGORITHM HM_WEIGHTED_DIRECTIONAL_FILTER	Weighted pixel average from different directions.

13.13.2.3 `_spinError`

enum `_spinError`

The error codes used in Spinnaker C.

These codes are returned from every function in Spinnaker C. 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	An error code of 0 means that the function has run without error.
SPINNAKER_ERR_ERROR	The error codes in the range of -1000 to -1999 are reserved for Spinnaker exceptions.
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	
SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT	The error codes in the range of -2000 to -2999 are reserved for Gen API related errors.
SPINNAKER_ERR_GENICAM_OUT_OF_RANGE	
SPINNAKER_ERR_GENICAM_PROPERTY	
SPINNAKER_ERR_GENICAM_RUN_TIME	
SPINNAKER_ERR_GENICAM_LOGICAL	
SPINNAKER_ERR_GENICAM_ACCESS	
SPINNAKER_ERR_GENICAM_TIMEOUT	
SPINNAKER_ERR_GENICAM_DYNAMIC_CAST	
SPINNAKER_ERR_GENICAM_GENERIC	
SPINNAKER_ERR_GENICAM_BAD_ALLOCATION	
SPINNAKER_ERR_IM_CONVERT	The error codes in the range of -3000 to -3999 are reserved for image processing related errors.
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	

Enumerator

SPINNAKER_ERR_CUSTOM_ID	Error codes less than -10000 are reserved for user-defined custom errors.

13.13.2.4 _spinImageFileFormat

enum `_spinImageFileFormat`

File formats to be used for saving images to disk.

Enumerator

SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT	Determine file format from file extension.
SPINNAKER_IMAGE_FILE_FORMAT_PGM	Portable gray map.
SPINNAKER_IMAGE_FILE_FORMAT_PPM	Portable pixmap.
SPINNAKER_IMAGE_FILE_FORMAT_BMP	Bitmap.
SPINNAKER_IMAGE_FILE_FORMAT_JPEG	JPEG.
SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000	JPEG 2000.
SPINNAKER_IMAGE_FILE_FORMAT_TIFF	Tagged image file format.
SPINNAKER_IMAGE_FILE_FORMAT_PNG	Portable network graphics.
SPINNAKER_IMAGE_FILE_FORMAT_RAW	Raw data.
SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS	

13.13.2.5 _spinImageStatus

enum `_spinImageStatus`

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

Enumerator

SPINNAKER_IMAGE_STATUS_UNKNOWN_ERR↔ OR	Image has an unknown error.
SPINNAKER_IMAGE_STATUS_NO_ERROR	Image is returned from <code>GetNextImage()</code> call without any errors.
SPINNAKER_IMAGE_STATUS_CRC_CHECK_F↔ AILED	Image failed CRC check.
SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW	Received more data than the size of the image.
SPINNAKER_IMAGE_STATUS_MISSING_PACK↔ ETS	Image has missing packets. Potential fixes include enabling jumbo packets and adjusting packet size/delay. For more information see https://www.flir.com/support-center/iis/machine-vision/application
SPINNAKER_IMAGE_STATUS_LEADER_BUFFE↔ R_SIZE_INCONSISTENT	Image leader is incomplete. Could be caused by missing packet(s). See link above.

Enumerator

SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT	Image trailer is incomplete. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT	Image has an inconsistent packet id. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_MISSING_LEADER	Image leader is missing. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_MISSING_TRAILER	Image trailer is missing. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE	Image data is incomplete. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_INFO_CORRUPTED	Image info is corrupted. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID	Image chunk data is invalid.
SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES	Image cannot be processed due to lack of system resources.

13.13.2.6 _spinLogLevel

enum `_spinLogLevel`

log levels

Enumerator

SPINNAKER_LOG_LEVEL_OFF	
SPINNAKER_LOG_LEVEL_FATAL	
SPINNAKER_LOG_LEVEL_ALERT	
SPINNAKER_LOG_LEVEL_CRIT	
SPINNAKER_LOG_LEVEL_ERROR	
SPINNAKER_LOG_LEVEL_WARN	
SPINNAKER_LOG_LEVEL_NOTICE	
SPINNAKER_LOG_LEVEL_INFO	
SPINNAKER_LOG_LEVEL_DEBUG	
SPINNAKER_LOG_LEVEL_NOTSET	

13.13.2.7 _spinStatisticsChannel

enum `_spinStatisticsChannel`

Channels that allow statistics to be calculated.

Enumerator

SPINNAKER_STATISTICS_CHANNEL_GREY	
SPINNAKER_STATISTICS_CHANNEL_RED	
SPINNAKER_STATISTICS_CHANNEL_GREEN	
SPINNAKER_STATISTICS_CHANNEL_BLUE	
SPINNAKER_STATISTICS_CHANNEL_HUE	
SPINNAKER_STATISTICS_CHANNEL_SATURATION	
SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS	
SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS	

13.13.2.8 _spinTIFFCompressionMethod

enum `_spinTIFFCompressionMethod`

Compression method to use for encoding TIFF images.

Enumerator

SPINNAKER_TIFF_COMPRESS_METHOD_NONE	
SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS	
SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE	
SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE	
SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3	
SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4	
SPINNAKER_TIFF_COMPRESS_METHOD_LZW	
SPINNAKER_TIFF_COMPRESS_METHOD_JPG	

13.13.2.9 _spinTLPayloadType

enum `_spinTLPayloadType`

Enumerator

SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN	
SPINNAKER_TLPAYLOAD_TYPE_IMAGE	
SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA	
SPINNAKER_TLPAYLOAD_TYPE_FILE	
SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA	
SPINNAKER_TLPAYLOAD_TYPE_JPEG	
SPINNAKER_TLPAYLOAD_TYPE_JPEG2000	
SPINNAKER_TLPAYLOAD_TYPE_H264	
SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY	
SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC	
SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART	

Enumerator

SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID	
SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED	
SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED	
SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED	

13.13.2.10 _spinTLPixelFormatNamespace

```
enum _spinTLPixelFormatNamespace
```

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

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

See also

[spinImageGetTLPixelFormat\(\)](#)

[spinImageGetTLPixelFormatNamespace\(\)](#)

Enumerator

SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT	
SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID	

13.13.3 Variable Documentation**13.13.3.1 False**

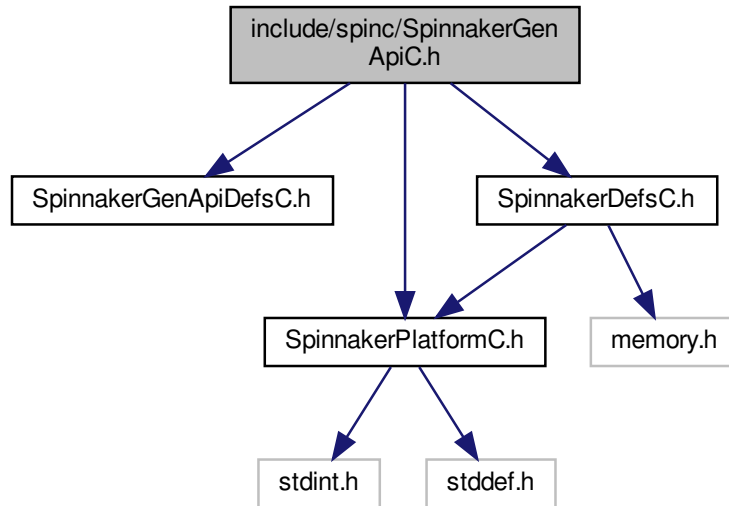
```
const bool8_t False = 0 [static]
```

13.13.3.2 True

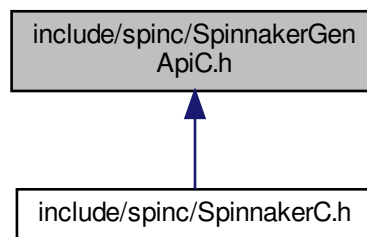
```
const bool8_t True = 1 [static]
```

13.14 include/spinc/SpinnakerGenApiC.h File Reference

Include dependency graph for SpinnakerGenApiC.h:



This graph shows which files directly or indirectly include this file:



Functions

- [SPINNAKERC_API spinNodeMapGetNode](#) ([spinNodeMapHandle](#) hNodeMap, const char *pName, [spinNodeHandle](#) *phNode)
Retrieves a node from the nodemap by name.
- [SPINNAKERC_API spinNodeMapGetNumNodes](#) ([spinNodeMapHandle](#) hNodeMap, size_t *pValue)
Gets the number of nodes in the map.
- [SPINNAKERC_API spinNodeMapGetNodeByIndex](#) ([spinNodeMapHandle](#) hNodeMap, size_t index, [spinNodeHandle](#) *phNode)

- Retrieves a node from the nodemap by index.*

 - [SPINNAKERC_API spinNodeMapReleaseNode](#) ([spinNodeMapHandle](#) hNodeMap, [spinNodeHandle](#) hNode)

Releases the entry node handle.
- [SPINNAKERC_API spinNodeMapPoll](#) ([spinNodeMapHandle](#) hNodeMap, [int64_t](#) timestamp)

Fires nodes which have a polling time.
- [SPINNAKERC_API spinNodesImplemented](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbResult)

Checks whether a node is implemented.
- [SPINNAKERC_API spinNodesReadable](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbResult)

Checks whether a node is readable.
- [SPINNAKERC_API spinNodesWritable](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbResult)

Checks whether a node is writable.
- [SPINNAKERC_API spinNodesAvailable](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbResult)

Checks whether a node is available.
- [SPINNAKERC_API spinNodesEqual](#) ([spinNodeHandle](#) hNodeFirst, [spinNodeHandle](#) hNodeSecond, [bool8_t](#) *pbResult)

Checks whether two nodes are equal.
- [SPINNAKERC_API spinNodeGetAccessMode](#) ([spinNodeHandle](#) hNode, [spinAccessMode](#) *pAccessMode)

Retrieves the access mode of a node (as an enum, spinAccessMode)
- [SPINNAKERC_API spinNodeGetName](#) ([spinNodeHandle](#) hNode, [char](#) *pBuf, [size_t](#) *pBufLen)

Retrieves the name of a node (no whitespace)
- [SPINNAKERC_API spinNodeGetNameSpace](#) ([spinNodeHandle](#) hNode, [spinNameSpace](#) *pNamespace)

Retrieve the namespace of a node (as an enum, spinNameSpace)
- [SPINNAKERC_API spinNodeGetVisibility](#) ([spinNodeHandle](#) hNode, [spinVisibility](#) *pVisibility)

Retrieves the recommended visibility of a node (as an enum, spinVisibility)
- [SPINNAKERC_API spinNodeInvalidateNode](#) ([spinNodeHandle](#) hNode)

Invalidates a node in case its values may have changed, rendering it no longer valid.
- [SPINNAKERC_API spinNodeGetCachingMode](#) ([spinNodeHandle](#) hNode, [spinCachingMode](#) *pCachingMode)

Retrieves the caching mode of a node (as an enum, spinCachingMode)
- [SPINNAKERC_API spinNodeGetToolTip](#) ([spinNodeHandle](#) hNode, [char](#) *pBuf, [size_t](#) *pBufLen)

Retrieves a short description of a node.
- [SPINNAKERC_API spinNodeGetDescription](#) ([spinNodeHandle](#) hNode, [char](#) *pBuf, [size_t](#) *pBufLen)

Retrieves a longer description of a node.
- [SPINNAKERC_API spinNodeGetDisplayName](#) ([spinNodeHandle](#) hNode, [char](#) *pBuf, [size_t](#) *pBufLen)

Retrieves the display name of a node (whitespace possible)
- [SPINNAKERC_API spinNodeGetType](#) ([spinNodeHandle](#) hNode, [spinNodeType](#) *pType)

Retrieves the type of a node (as an enum, spinNodeType)
- [SPINNAKERC_API spinNodeGetPollingTime](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pPollingTime)

Retrieve the polling time of a node.
- [SPINNAKERC_API spinNodeRegisterCallback](#) ([spinNodeHandle](#) hNode, [spinNodeCallbackFunction](#) pCbFunction, [spinNodeCallbackHandle](#) *phCb)

Registers a callback to a node.
- [SPINNAKERC_API spinNodeDeregisterCallback](#) ([spinNodeHandle](#) hNode, [spinNodeCallbackHandle](#) hCb)

Unregisters a callback from a node.
- [SPINNAKERC_API spinNodeGetImposedAccessMode](#) ([spinNodeHandle](#) hNode, [spinAccessMode](#) imposedAccessMode)

Retrieves the imposed access mode of a node.
- [SPINNAKERC_API spinNodeGetImposedVisibility](#) ([spinNodeHandle](#) hNode, [spinVisibility](#) imposedVisibility)

Retrieves the imposed visibility of a node.
- [SPINNAKERC_API spinNodeToString](#) ([spinNodeHandle](#) hNode, [char](#) *pBuf, [size_t](#) *pBufLen)

Retrieves the value of any node type as a c-string.

- [SPINNAKERC_API spinNodeToStringEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, char *pBuf, [size_t](#) *pBufLen)

Retrieves the value of any node type as a c-string; manually set whether to verify the node.
- [SPINNAKERC_API spinNodeFromString](#) ([spinNodeHandle](#) hNode, const char *pBuf)

Sets the value of any node type from a c-string; it is important to ensure that the value of the c-string is appropriate to the node type.
- [SPINNAKERC_API spinNodeFromStringEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, const char *pBuf)

Sets the value of any node type from a c-string; manually set whether to verify the node; ensure the value of the c-string is appropriate to the node type.
- [SPINNAKERC_API spinStringSetValue](#) ([spinNodeHandle](#) hNode, const char *pBuf)

Sets the value of a string node.
- [SPINNAKERC_API spinStringSetValueEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, const char *pBuf)

Sets the value of a string node; manually set whether to verify the node.
- [SPINNAKERC_API spinStringGetValue](#) ([spinNodeHandle](#) hNode, char *pBuf, [size_t](#) *pBufLen)

Retrieves the value of a string node as a c-string.
- [SPINNAKERC_API spinStringGetValueEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, char *pBuf, [size_t](#) *pBufLen)

Retrieves the value of a string node as a cstring; manually set whether to verify the node.
- [SPINNAKERC_API spinStringGetMaxLength](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pValue)

Retrieves the maximum length of the c-string to be returned.
- [SPINNAKERC_API spinIntegerSetValue](#) ([spinNodeHandle](#) hNode, [int64_t](#) value)

Sets the value of an integer node.
- [SPINNAKERC_API spinIntegerSetValueEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, [int64_t](#) value)

Sets the value of an integer node; manually set whether to verify the node.
- [SPINNAKERC_API spinIntegerGetValue](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pValue)

Retrieves the value of an integer node.
- [SPINNAKERC_API spinIntegerGetValueEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, [int64_t](#) *pValue)

Retrieves the value of an integer node; manually set whether to verify the node.
- [SPINNAKERC_API spinIntegerGetMin](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pValue)

Retrieves the minimum value of an integer node; all potential values must be greater than or equal to the minimum.
- [SPINNAKERC_API spinIntegerGetMax](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pValue)

Retrieves the maximum value of an integer node; all potential values must be lesser than or equal to the maximum.
- [SPINNAKERC_API spinIntegerGetInc](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pValue)

Retrieves the increment of an integer node; all possible values must be divisible by the increment.
- [SPINNAKERC_API spinIntegerGetRepresentation](#) ([spinNodeHandle](#) hNode, [spinRepresentation](#) *pValue)

Retrieves the numerical representation of the value of a node; i.e.
- [SPINNAKERC_API spinFloatSetValue](#) ([spinNodeHandle](#) hNode, double value)

Sets the value of a float node.
- [SPINNAKERC_API spinFloatSetValueEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, double value)

Sets the value of a float node; manually set whether to verify the node.
- [SPINNAKERC_API spinFloatGetValue](#) ([spinNodeHandle](#) hNode, double *pValue)

Retrieves the value of a float node.
- [SPINNAKERC_API spinFloatGetValueEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, double *pValue)

Retrieves the value of a float node; manually set whether to verify the node.
- [SPINNAKERC_API spinFloatGetMin](#) ([spinNodeHandle](#) hNode, double *pValue)

Retrieves the minimum value of a float node; all potential values must be greater than or equal to the minimum.
- [SPINNAKERC_API spinFloatGetMax](#) ([spinNodeHandle](#) hNode, double *pValue)

Retrieves the maximum value of a float node; all potential values must be lesser than or equal to the maximum.
- [SPINNAKERC_API spinFloatGetRepresentation](#) ([spinNodeHandle](#) hNode, [spinRepresentation](#) *pValue)

Retrieves the numerical representation of the value of a node; i.e.
- [SPINNAKERC_API spinFloatGetUnit](#) ([spinNodeHandle](#) hNode, char *pBuf, [size_t](#) *pBufLen)

- Retrieves the units of the float node value.*

 - [SPINNAKERC_API spinEnumerationGetNumEntries](#) ([spinNodeHandle](#) hEnumNode, [size_t](#) *pValue)

Retrieves the number of entries of an enum node.

 - [SPINNAKERC_API spinEnumerationGetEntryByIndex](#) ([spinNodeHandle](#) hEnumNode, [size_t](#) index, [spinNodeHandle](#) *phEntry)

Retrieves an entry node from an enum node using an index.

 - [SPINNAKERC_API spinEnumerationGetEntryByName](#) ([spinNodeHandle](#) hEnumNode, [const char](#) *pName, [spinNodeHandle](#) *phEntry)

Retrieves an entry node from an enum node using the entry's symbolic.

 - [SPINNAKERC_API spinEnumerationGetCurrentEntry](#) ([spinNodeHandle](#) hEnumNode, [spinNodeHandle](#) *phEntry)

Retrieves the currently selected entry node from an enum node.

 - [SPINNAKERC_API spinEnumerationReleaseNode](#) ([spinNodeHandle](#) hEnumNode, [spinNodeHandle](#) hEntry)

Releases the entry node from the enum node handle.

 - [SPINNAKERC_API spinEnumerationSetIntValue](#) ([spinNodeHandle](#) hEnumNode, [int64_t](#) value)

Sets a new entry using its integer value retrieved from a call to [spinEnumerationEntryGetIntValue\(\)](#); note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

 - [SPINNAKERC_API spinEnumerationSetEnumValue](#) ([spinNodeHandle](#) hEnumNode, [size_t](#) value)

Sets a new entry using its enum; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

 - [SPINNAKERC_API spinEnumerationEntryGetIntValue](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pValue)

Retrieves the integer value of an entry node; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

 - [SPINNAKERC_API spinEnumerationEntryGetEnumValue](#) ([spinNodeHandle](#) hNode, [size_t](#) *pValue)

Retrieves the enum value (as an integer) of an entry node; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

 - [SPINNAKERC_API spinEnumerationEntryGetSymbolic](#) ([spinNodeHandle](#) hNode, [char](#) *pBuf, [size_t](#) *pBufLen)

Retrieves the symbolic of an entry node as a c-string.

 - [SPINNAKERC_API spinBooleanSetValue](#) ([spinNodeHandle](#) hNode, [bool8_t](#) value)

Sets the value of a boolean node; boolean values are represented by 'True' (which equals '0') and 'False' (which equals '1')

 - [SPINNAKERC_API spinBooleanGetValue](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbValue)

Retrieves the value of a boolean node; boolean values are represented by 'True' (which equals '0') and 'False' (which equals '1')

 - [SPINNAKERC_API spinCommandExecute](#) ([spinNodeHandle](#) hNode)

Executes the action associated to a command node.

 - [SPINNAKERC_API spinCommandIsDone](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbValue)

Retrieves whether or not the action of a command node has completed.

 - [SPINNAKERC_API spinCategoryGetNumFeatures](#) ([spinNodeHandle](#) hCategoryNode, [size_t](#) *pValue)

Retrieves the number of a features (or child nodes) of a category node.

 - [SPINNAKERC_API spinCategoryGetFeatureByIndex](#) ([spinNodeHandle](#) hCategoryNode, [size_t](#) index, [spinNodeHandle](#) *phFeature)

Retrieves a node from a category node using an index.

 - [SPINNAKERC_API spinCategoryReleaseNode](#) ([spinNodeHandle](#) hCategoryNode, [spinNodeHandle](#) hFeature)

Releases the feature node from the category node.

 - [SPINNAKERC_API spinRegisterGet](#) ([spinNodeHandle](#) hNode, [uint8_t](#) *pBuf, [int64_t](#) length)

Retrieves the value of a register node.

 - [SPINNAKERC_API spinRegisterGetEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, [bool8_t](#) bIgnoreCache, [uint8_t](#) *pBuf, [int64_t](#) length)

Retrieves the value of a register node; manually set whether to verify the node and whether to ignore the cache.

 - [SPINNAKERC_API spinRegisterGetAddress](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pAddress)

Retrieves the address of a register node.

- [SPINNAKERC_API spinRegisterGetLength](#) ([spinNodeHandle](#) hNode, [int64_t](#) *pLength)

Retrieves the length (in bytes) of the value of a register node.

- [SPINNAKERC_API spinRegisterSet](#) ([spinNodeHandle](#) hNode, [const uint8_t](#) *pBuf, [int64_t](#) length)

Sets the value of a register node.

- [SPINNAKERC_API spinRegisterSetEx](#) ([spinNodeHandle](#) hNode, [bool8_t](#) bVerify, [const uint8_t](#) *pBuf, [int64_t](#) length)

Sets the value of a register node; manually set whether to verify the node.

- [SPINNAKERC_API spinRegisterSetReference](#) ([spinNodeHandle](#) hNode, [spinNodeHandle](#) hRef)

Uses a second node as a reference for a register node.

13.14.1 Function Documentation

13.14.1.1 spinBooleanGetValue()

```
SPINNAKERC_API spinBooleanGetValue (
    spinNodeHandle hNode,
    bool8_t * pbValue )
```

Retrieves the value of a boolean node; boolean values are represented by 'True' (which equals '0') and 'False' (which equals '1')

See also

[spinError](#)

Parameters

<i>hNode</i>	The boolean node of the value to read
<i>pValue</i>	The boolean pointer in which the value is returned

Returns

[spinError](#) The error code; returns [SPINNAKER_ERR_SUCCESS](#) (or 0) for no error

13.14.1.2 spinBooleanSetValue()

```
SPINNAKERC_API spinBooleanSetValue (
    spinNodeHandle hNode,
    bool8_t value )
```

Sets the value of a boolean node; boolean values are represented by 'True' (which equals '0') and 'False' (which equals '1')

See also

[spinError](#)

Parameters

<i>hNode</i>	The boolean node having its value changed
<i>value</i>	The boolean value to set

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.3 spinCategoryGetFeatureByIndex()

```
SPINNAKERC_API spinCategoryGetFeatureByIndex (
    spinNodeHandle hCategoryNode,
    size_t index,
    spinNodeHandle * phFeature )
```

Retrieves a node from a category node using an index.

See also

spinError

Parameters

<i>hCategoryNode</i>	The category node of the node to retrieve
<i>index</i>	The index of the feature node
<i>phFeature</i>	The node handle pointer in which the feature node is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.4 spinCategoryGetNumFeatures()

```
SPINNAKERC_API spinCategoryGetNumFeatures (
    spinNodeHandle hCategoryNode,
    size_t * pValue )
```

Retrieves the number of a features (or child nodes) or a category node.

See also

spinError

Parameters

<i>hCategoryNode</i>	The category node where the features to be counted are
<i>pValue</i>	The unsigned integer pointer in which the number of features is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.5 spinCategoryReleaseNode()

```
SPINNAKERC_API spinCategoryReleaseNode (
    spinNodeHandle hCategoryNode,
    spinNodeHandle hFeature )
```

Releases the feature node from the category node.

Make sure node handle is cleaned up properly by setting it to NULL after the node is released. If this function is not explicitly called, the handle will be released upon the release of the camera handle.

See also

[spinCameraRelease](#)

`spinError`

Parameters

<i>hCategoryNode</i>	The category node handle from which the feature node is retrieved
<i>hFeature</i>	The feature node handle to be released

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.6 spinCommandExecute()

```
SPINNAKERC_API spinCommandExecute (
    spinNodeHandle hNode )
```

Executes the action associated to a command node.

See also

`spinError`

Parameters

<i>hNode</i>	The command node to execute

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.7 spinCommandIsDone()

```
SPINNAKERC_API spinCommandIsDone (
    spinNodeHandle hNode,
    bool8_t * pbValue )
```

Retrieves whether or not the action of a command node has completed.

See also

spinError

Parameters

<i>hNode</i>	The command node to check
<i>pValue</i>	The boolean pointer to return whether or not the command has completed

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.8 spinEnumerationEntryGetEnumValue()

```
SPINNAKERC_API spinEnumerationEntryGetEnumValue (
    spinNodeHandle hNode,
    size_t * pValue )
```

Retrieves the enum value (as an integer) of an entry node; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

See also

[spinEnumerationSetEnumValue\(\)](#)

spinError

Parameters

<i>hNode</i>	The entry node of the enum value to retrieve
<i>pValue</i>	The unsigned integer pointer in which the enum value of the entry is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.9 spinEnumerationEntryGetIntValue()

```
SPINNAKERC_API spinEnumerationEntryGetIntValue (
    spinNodeHandle hNode,
    int64_t * pValue )
```

Retrieves the integer value of an entry node; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

See also

[spinEnumerationSetIntValue\(\)](#)

spinError

Parameters

<i>hNode</i>	The entry node of the integer value to retrieve
<i>pValue</i>	The integer pointer in which the integer value of the entry is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.10 spinEnumerationEntryGetSymbolic()

```
SPINNAKERC_API spinEnumerationEntryGetSymbolic (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the symbolic of an entry node as a c-string.

See also

spinError

Parameters

<i>hNode</i>	The entry node of the symbolic to retrieve
<i>pBuf</i>	The c-string character buffer in which the symbolic of the entry node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.11 spinEnumerationGetCurrentEntry()

```
SPINNAKERC_API spinEnumerationGetCurrentEntry (
    spinNodeHandle hEnumNode,
    spinNodeHandle * phEntry )
```

Retrieves the currently selected entry node from an enum node.

See also

spinError

Parameters

<i>hEnumNode</i>	The enum node from which the current entry node is retrieved
<i>phEntry</i>	The node handle pointer in which the current entry node is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.12 spinEnumerationGetEntryByIndex()

```
SPINNAKERC_API spinEnumerationGetEntryByIndex (
    spinNodeHandle hEnumNode,
    size_t index,
    spinNodeHandle * phEntry )
```

Retrieves an entry node from an enum node using an index.

See also

spinError

Parameters

<i>hEnumNode</i>	The enum node from which the entry node is retrieved
<i>index</i>	The index of the entry node
<i>phEntry</i>	The node handle pointer in which the entry node is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.13 spinEnumerationGetEntryByName()

```
SPINNAKERC_API spinEnumerationGetEntryByName (
    spinNodeHandle hEnumNode,
    const char * pName,
    spinNodeHandle * phEntry )
```

Retrieves an entry node from an enum node using the entry's symbolic.

See also

spinError

Parameters

<i>hEnumNode</i>	The enum node from which the entry node is retrieved
<i>pName</i>	The name of the entry node
<i>phEntry</i>	The node handle pointer in which the entry node is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.14 spinEnumerationGetNumEntries()

```
SPINNAKERC_API spinEnumerationGetNumEntries (
    spinNodeHandle hEnumNode,
    size_t * pValue )
```

Retrieves the number of entries of an enum node.

See also

spinError

Parameters

<i>hEnumNode</i>	The enum node where the entries to be counted are
<i>pValue</i>	The unsigned integer pointer in which the number of entries is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.15 spinEnumerationReleaseNode()

```
SPINNAKERC_API spinEnumerationReleaseNode (  
    spinNodeHandle hEnumNode,  
    spinNodeHandle hEntry )
```

Releases the entry node from the enum node handle.

Make sure node handle is cleaned up properly by setting it to NULL after the node is released. If this function is not explicitly called, the handle will be released upon the release of the camera handle.

See also

[spinCameraRelease](#)

`spinError`

Parameters

<i>hEnumNode</i>	The enum node from which the current entry node is retrieved
<i>hEntry</i>	The entry node handle to be released

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.16 spinEnumerationSetEnumValue()

```
SPINNAKERC_API spinEnumerationSetEnumValue (  
    spinNodeHandle hEnumNode,  
    size_t value )
```

Sets a new entry using its enum; note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

See also

[spinEnumerationEntryGetEnumValue\(\)](#)

`spinError`

Parameters

<i>hEnumNode</i>	The enum node have its entry changed
<i>value</i>	The enum value of the entry node to set; this corresponds to its integer value created in the library

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.17 spinEnumerationSetIntValue()

```
SPINNAKERC_API spinEnumerationSetIntValue (
    spinNodeHandle hEnumNode,
    int64_t value )
```

Sets a new entry using its integer value retrieved from a call to [spinEnumerationEntryGetIntValue\(\)](#); note that enumeration entry int and enum values are different - int values defined on camera, enum values found in [SpinnakerDefsC.h](#).

See also

[spinEnumerationEntryGetIntValue\(\)](#)
`spinError`

Parameters

<i>hEnumNode</i>	The enum node having its entry changed
<i>value</i>	The integer value of the entry node to set; this corresponds to the integer value internal to the camera

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.18 spinFloatGetMax()

```
SPINNAKERC_API spinFloatGetMax (
    spinNodeHandle hNode,
    double * pValue )
```

Retrieves the maximum value of a float node; all potential values must be lesser than or equal to the maximum.

See also

`spinError`

Parameters

<i>hNode</i>	The float node of the maximum value to retrieve
<i>pValue</i>	The double pointer in which the maximum value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.19 spinFloatGetMin()

```
SPINNAKERC_API spinFloatGetMin (  
    spinNodeHandle hNode,  
    double * pValue )
```

Retrieves the minimum value of a float node; all potential values must be greater than or equal to the minimum.

See also

spinError

Parameters

<i>hNode</i>	The float node of the minimum value to retrieve
<i>pValue</i>	The double pointer in which the minimum value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.20 spinFloatGetRepresentation()

```
SPINNAKERC_API spinFloatGetRepresentation (  
    spinNodeHandle hNode,  
    spinRepresentation * pValue )
```

Retrieves the numerical representation of the value of a node; i.e.

linear, logarithmic, hexadecimal, MAC address, etc.

See also

spinError

Parameters

<i>hNode</i>	The float node of the numerical representation to retrieve
<i>pValue</i>	The representation enum pointer in which the type of numerical representation is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.21 spinFloatGetUnit()

```
SPINNAKERC_API spinFloatGetUnit (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the units of the float node value.

See also

`spinError`

Parameters

<i>hNode</i>	The float node of the units to retrieve
<i>pBuf</i>	The c-string character buffer in which the value units are returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.22 spinFloatGetValue()

```
SPINNAKERC_API spinFloatGetValue (
    spinNodeHandle hNode,
    double * pValue )
```

Retrieves the value of a float node.

See also

`spinError`

Parameters

<i>hNode</i>	The float node of the value to read
<i>pValue</i>	The double pointer in which the value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.23 spinFloatGetValueEx()

```
SPINNAKERC_API spinFloatGetValueEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    double * pValue )
```

Retrieves the value of a float node; manually set whether to verify the node.

See also

spinError

Parameters

<i>hNode</i>	The float node of the value to read
<i>pValue</i>	The double pointer in which the value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.24 spinFloatSetValue()

```
SPINNAKERC_API spinFloatSetValue (
    spinNodeHandle hNode,
    double value )
```

Sets the value of a float node.

See also

spinError

Parameters

<i>hNode</i>	The float node having its value changed
<i>value</i>	The float value to set

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.25 spinFloatSetValueEx()

```
SPINNAKERC_API spinFloatSetValueEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    double value )
```

Sets the value of a float node; manually set whether to verify the node.

See also

`spinError`

Parameters

<i>hNode</i>	The float node having its value changed
<i>bVerify</i>	The boolean of whether to verify the node
<i>value</i>	The float value to set

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.26 spinIntegerGetInc()

```
SPINNAKERC_API spinIntegerGetInc (
    spinNodeHandle hNode,
    int64_t * pValue )
```

Retrieves the increment of an integer node; all possible values must be divisible by the increment.

See also

`spinError`

Parameters

<i>hNode</i>	The integer node of the increment to retrieve
<i>pValue</i>	The integer pointer in which the increment is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.27 spinIntegerGetMax()

```
SPINNAKERC_API spinIntegerGetMax (  
    spinNodeHandle hNode,  
    int64_t * pValue )
```

Retrieves the maximum value of an integer node; all potential values must be lesser than or equal to the maximum.

See also

spinError

Parameters

<i>hNode</i>	The integer node of the maximum value to retrieve
<i>pValue</i>	The integer pointer in which the maximum value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.28 spinIntegerGetMin()

```
SPINNAKERC_API spinIntegerGetMin (  
    spinNodeHandle hNode,  
    int64_t * pValue )
```

Retrieves the minimum value of an integer node; all potential values must be greater than or equal to the minimum.

See also

spinError

Parameters

<i>hNode</i>	The integer node of the minimum value to retrieve
<i>pValue</i>	The integer pointer in which the minimum value is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.29 spinIntegerGetRepresentation()

```
SPINNAKERC_API spinIntegerGetRepresentation (
    spinNodeHandle hNode,
    spinRepresentation * pValue )
```

Retrieves the numerical representation of the value of a node; i.e.

linear, logarithmic, hexadecimal, MAC address, etc.

See also

`spinError`

Parameters

<i>hNode</i>	The integer node of the numerical representation to retrieve
<i>pValue</i>	The representation enum pointer in which the type of numerical representation is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.30 spinIntegerGetValue()

```
SPINNAKERC_API spinIntegerGetValue (
    spinNodeHandle hNode,
    int64_t * pValue )
```

Retrieves the value of an integer node.

See also

`spinError`

Parameters

<i>hNode</i>	The integer node of the value to read
<i>pValue</i>	The integer pointer in which the value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.31 spinIntegerGetValueEx()

```
SPINNAKERC_API spinIntegerGetValueEx (  
    spinNodeHandle hNode,  
    bool8_t bVerify,  
    int64_t * pValue )
```

Retrieves the value of an integer node; manually set whether to verify the node.

See also

spinError

Parameters

<i>hNode</i>	The integer node of the value to read
<i>bVerify</i>	The boolean of whether to verify the node
<i>pValue</i>	The integer pointer in which the value is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.32 spinIntegerSetValue()

```
SPINNAKERC_API spinIntegerSetValue (  
    spinNodeHandle hNode,  
    int64_t value )
```

Sets the value of an integer node.

See also

spinError

Parameters

<i>hNode</i>	The integer node having its value changed
<i>value</i>	The integer value to set

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.33 spinIntegerSetValueEx()

```
SPINNAKERC_API spinIntegerSetValueEx (  
    spinNodeHandle hNode,  
    bool8_t bVerify,  
    int64_t value )
```

Sets the value of an integer node; manually set whether to verify the node.

See also

`spinError`

Parameters

<i>hNode</i>	The integer node having its value changed
<i>bVerify</i>	The boolean of whether to verify the node
<i>value</i>	The integer value to set

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.34 spinNodeDeregisterCallback()

```
SPINNAKERC_API spinNodeDeregisterCallback (  
    spinNodeHandle hNode,  
    spinNodeCallbackHandle hCb )
```

Unregisters a callback from a node.

See also

`spinError`

Parameters

<i>hNode</i>	The node from which to unregister the callback
<i>hCb</i>	The callback handle to unregister

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.35 spinNodeFromString()

```
SPINNAKERC_API spinNodeFromString (
    spinNodeHandle hNode,
    const char * pBuf )
```

Sets the value of any node type from a c-string; it is important to ensure that the value of the c-string is appropriate to the node type.

See also

spinError

Parameters

<i>hNode</i>	The node having its value changed
<i>pBuf</i>	The c-string of the value to set

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.36 spinNodeFromStringEx()

```
SPINNAKERC_API spinNodeFromStringEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    const char * pBuf )
```

Sets the value of any node type from a c-string; manually set whether to verify the node; ensure the value of the c-string is appropriate to the node type.

See also

spinError

Parameters

<i>hNode</i>	The node having its value changed
<i>bVerify</i>	The boolean of whether to verify the node
<i>pBuf</i>	The c-string of the value to set

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.37 spinNodeGetAccessMode()

```
SPINNAKERC_API spinNodeGetAccessMode (
    spinNodeHandle hNode,
    spinAccessMode * pAccessMode )
```

Retrieves the access mode of a node (as an enum, spinAccessMode)

See also

spinError
spinAccessMode

Parameters

<i>hNode</i>	The node of the access mode to retrieve
<i>pAccessMode</i>	The access mode enum pointer in which the access mode is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.38 spinNodeGetCachingMode()

```
SPINNAKERC_API spinNodeGetCachingMode (
    spinNodeHandle hNode,
    spinCachingMode * pCachingMode )
```

Retrieves the caching mode of a node (as an enum, spinCachingMode)

See also

spinError
spinCachingMode

Parameters

<i>hNode</i>	The node of the caching mode to retrieve
<i>pCachingMode</i>	The caching mode enum pointer in which the caching mode is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.39 spinNodeGetDescription()

```
SPINNAKERC_API spinNodeGetDescription (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves a longer description of a node.

See also

spinError

Parameters

<i>hNode</i>	The node of the description to retrieve
<i>pBuf</i>	The c-string character buffer in which the longer description of the node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.40 spinNodeGetDisplayName()

```
SPINNAKERC_API spinNodeGetDisplayName (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the display name of a node (whitespace possible)

See also

spinError

Parameters

<i>hNode</i>	The node of the display name to retrieve
<i>pBuf</i>	The c-string character buffer in which the display name of the node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.41 spinNodeGetImposedAccessMode()

```
SPINNAKERC_API spinNodeGetImposedAccessMode (
    spinNodeHandle hNode,
    spinAccessMode imposedAccessMode )
```

Retrieves the imposed access mode of a node.

See also

`spinError`

Parameters

<i>hNode</i>	The node of the imposed access mode to retrieve
<i>imposedAccessMode</i>	The access mode enum pointer in which the imposed access mode is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.42 spinNodeGetImposedVisibility()

```
SPINNAKERC_API spinNodeGetImposedVisibility (
    spinNodeHandle hNode,
    spinVisibility imposedVisibility )
```

Retrieves the imposed visibility of a node.

See also

`spinError`

Parameters

<i>hNode</i>	The node of the visibility to impose
<i>imposedVisibility</i>	The visibility enum pointer in which the imposed visibility is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.43 spinNodeGetName()

```
SPINNAKERC_API spinNodeGetName (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the name of a node (no whitespace)

See also

spinError

Parameters

<i>hNode</i>	The node of the name to retrieve
<i>pBuf</i>	The c-string character buffer in which the name of the node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.44 spinNodeGetNameSpace()

```
SPINNAKERC_API spinNodeGetNameSpace (
    spinNodeHandle hNode,
    spinNameSpace * pNamespace )
```

Retrieve the namespace of a node (as an enum, spinNameSpace)

See also

spinError
spinNameSpace

Parameters

<i>hNode</i>	The node of the namespace to retrieve
<i>pNamespace</i>	The namespace enum pointer in which the namespace is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.45 spinNodeGetPollingTime()

```
SPINNAKERC_API spinNodeGetPollingTime (
    spinNodeHandle hNode,
    int64_t * pPollingTime )
```

Retrieve the polling time of a node.

See also

spinError

Parameters

<i>hNode</i>	The node of the polling time to retrieve
<i>pPollingTime</i>	The integer pointer in which the polling time is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.46 spinNodeGetToolTip()

```
SPINNAKERC_API spinNodeGetToolTip (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves a short description of a node.

See also

spinError

Parameters

<i>hNode</i>	The node of the tooltip to retrieve
<i>pBuf</i>	The c-string character buffer in which the short description of the node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.47 spinNodeGetType()

```
SPINNAKERC_API spinNodeGetType (
    spinNodeHandle hNode,
    spinNodeType * pType )
```

Retrieves the type of a node (as an enum, spinNodeType)

See also

spinError
spinNodeType

Parameters

<i>hNode</i>	The node of the node type to retrieve
<i>pType</i>	The node type enum pointer in which the type of node is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.48 spinNodeGetVisibility()

```
SPINNAKERC_API spinNodeGetVisibility (
    spinNodeHandle hNode,
    spinVisibility * pVisibility )
```

Retrieves the recommended visibility of a node (as an enum, spinVisibility)

See also

spinError
spinVisibility

Parameters

<i>hNode</i>	The node of the visibility to retrieve
<i>pVisibility</i>	The visibility enum pointer in which the visibility is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.49 spinNodeInvalidateNode()

```
SPINNAKERC_API spinNodeInvalidateNode (
    spinNodeHandle hNode )
```

Invalidates a node in case its values may have changed, rendering it no longer valid.

See also

spinError

Parameters

<i>hNode</i>	The node whose values may have changed

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.50 spinNodesAvailable()

```
SPINNAKERC_API spinNodeIsAvailable (
    spinNodeHandle hNode,
    bool8_t * pbResult )
```

Checks whether a node is available.

See also

spinError

Parameters

<i>hNode</i>	The node to check
<i>pbResult</i>	The boolean pointer to return whether or not the node is available

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.51 spinNodeIsEqual()

```
SPINNAKERC_API spinNodeIsEqual (
    spinNodeHandle hNodeFirst,
    spinNodeHandle hNodeSecond,
    bool8_t * pbResult )
```

Checks whether two nodes are equal.

See also

spinError

Parameters

<i>hNodeFirst</i>	The first node to check
<i>hNodeSecond</i>	The second node to check
<i>pbResult</i>	The boolean pointer to return whether or not the two nodes are equal

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.52 spinNodeIsImplemented()

```
SPINNAKERC_API spinNodeIsImplemented (
    spinNodeHandle hNode,
    bool8_t * pbResult )
```

Checks whether a node is implemented.

See also

spinError

Parameters

<i>hNode</i>	The node to check
<i>pbResult</i>	The boolean pointer to return whether or not the node is implemented

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.53 spinNodeIsReadable()

```
SPINNAKERC_API spinNodeIsReadable (
    spinNodeHandle hNode,
    bool8_t * pbResult )
```

Checks whether a node is readable.

See also

spinError

Parameters

<i>hNode</i>	The node to check
<i>pbResult</i>	The boolean pointer to return whether or not the node is readable

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.54 spinNodeIsWritable()

```
SPINNAKERC_API spinNodeIsWritable (
    spinNodeHandle hNode,
    bool8_t * pbResult )
```

Checks whether a node is writable.

See also

spinError

Parameters

<i>hNode</i>	The node to check
<i>pbResult</i>	The boolean pointer to return whether or not the node is writable

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.55 spinNodeMapGetNode()

```
SPINNAKERC_API spinNodeMapGetNode (
    spinNodeMapHandle hNodeMap,
    const char * pName,
    spinNodeHandle * pNode )
```

Retrieves a node from the nodemap by name.

See also

spinError

Parameters

<i>hNodeMap</i>	The node map where the node is
<i>pName</i>	The name of the node
<i>pNode</i>	The node handle pointer in which the node is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.56 spinNodeMapGetNodeByIndex()

```
SPINNAKERC_API spinNodeMapGetNodeByIndex (
    spinNodeMapHandle hNodeMap,
    size_t index,
    spinNodeHandle * pNode )
```

Retrieves a node from the nodemap by index.

See also

spinError

Parameters

<i>hNodeMap</i>	The node map where the node is
<i>index</i>	The index of the node
<i>pNode</i>	The node handle pointer in which the node is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.57 spinNodeMapGetNumNodes()

```
SPINNAKERC_API spinNodeMapGetNumNodes (
    spinNodeMapHandle hNodeMap,
    size_t * pValue )
```

Gets the number of nodes in the map.

See also

`spinError`

Parameters

<i>hNodeMap</i>	The node map where the nodes to be counted are
<i>pValue</i>	The unsigned integer pointer in which the number of nodes is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.58 spinNodeMapPoll()

```
SPINNAKERC_API spinNodeMapPoll (
    spinNodeMapHandle hNodeMap,
    int64_t timestamp )
```

Fires nodes which have a polling time.

See also

`spinError`

Parameters

<i>hNodeMap</i>	The nodemap to poll
<i>timestamp</i>	The timestamp

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.59 spinNodeMapReleaseNode()

```
SPINNAKERC_API spinNodeMapReleaseNode (
    spinNodeMapHandle hNodeMap,
    spinNodeHandle hNode )
```

Releases the entry node handle.

Make sure node handle is cleaned up properly by setting it to NULL after the node is released. If this function is not explicitly called, the handle will be released upon the release of the camera handle.

See also

[spinCameraRelease](#)

spinError

Parameters

<i>hNodeMap</i>	The node map from which the node handle is retrieved
<i>hNode</i>	The node handle to be released

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.60 spinNodeRegisterCallback()

```
SPINNAKERC_API spinNodeRegisterCallback (
    spinNodeHandle hNode,
    spinNodeCallbackFunction pCbFunction,
    spinNodeCallbackHandle * phCb )
```

Registers a callback to a node.

See also

spinError

Parameters

<i>hNode</i>	The node on which to register the callback
<i>pCbFunction</i>	The function pointer of the function that will execute when the callback is triggered; must match signature "void spinNodeCallbackFunction(spinNodeHandle hNode)"
<i>phCb</i>	The callback handle pointer in which the callback is returned; used to unregister callbacks

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.61 spinNodeToString()

```
SPINNAKER_API spinNodeToString (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the value of any node type as a c-string.

See also

`spinError`

Parameters

<i>hNode</i>	The node of the value to read
<i>pBuf</i>	The c-string character buffer in which the value of the node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.62 spinNodeToStringEx()

```
SPINNAKER_API spinNodeToStringEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the value of any node type as a c-string; manually set whether to verify the node.

See also

`spinError`

Parameters

<i>hNode</i>	The node of the value to read
<i>bVerify</i>	The boolean of whether to verify the node
<i>pBuf</i>	The c-string character buffer in which the value of the node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.63 spinRegisterGet()

```
SPINNAKERC_API spinRegisterGet (
    spinNodeHandle hNode,
    uint8_t * pBuf,
    int64_t length )
```

Retrieves the value of a register node.

See also

spinError

Parameters

<i>hNode</i>	The register node of the value to retrieve
<i>pBuf</i>	The unsigned integer buffer in which the value is returned
<i>length</i>	The integer pointer in which the length of the register array is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.64 spinRegisterGetAddress()

```
SPINNAKERC_API spinRegisterGetAddress (
    spinNodeHandle hNode,
    int64_t * pAddress )
```

Retrieves the address of a register node.

See also

spinError

Parameters

<i>hNode</i>	The register node of the address to retrieve
<i>pAddress</i>	The integer pointer in which the address is returned

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.65 spinRegisterGetEx()

```
SPINNAKERC_API spinRegisterGetEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    bool8_t bIgnoreCache,
    uint8_t * pBuf,
    int64_t length )
```

Retrieves the value of a register node; manually set whether to verify the node and whether to ignore the cache.

See also

`spinError`

Parameters

<i>hNode</i>	The register node of the value to retrieve
<i>bVerify</i>	The boolean of whether to verify the node
<i>IgnoreCache</i>	The boolean of whether to ignore the cache
<i>pBuf</i>	The unsigned integer buffer in which the value is returned
<i>length</i>	The integer pointer in which the length of the register array is returned; the input value is the maximum length

Returns

`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

13.14.1.66 spinRegisterGetLength()

```
SPINNAKERC_API spinRegisterGetLength (
    spinNodeHandle hNode,
    int64_t * pLength )
```

Retrieves the length (in bytes) of the value of a register node.

See also

`spinError`

Parameters

<i>hNode</i>	The register node of the length to retrieve
<i>pLength</i>	The integer in which the number of bytes is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.67 spinRegisterSet()

```
SPINNAKERC_API spinRegisterSet (  
    spinNodeHandle hNode,  
    const uint8_t * pBuf,  
    int64_t length )
```

Sets the value of a register node.

See also

spinError

Parameters

<i>hNode</i>	The register node of the value to set
<i>pBuf</i>	The unsigned integer buffer of the value to set
<i>length</i>	The number of bytes of the value to set

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.68 spinRegisterSetEx()

```
SPINNAKERC_API spinRegisterSetEx (  
    spinNodeHandle hNode,  
    bool8_t bVerify,  
    const uint8_t * pBuf,  
    int64_t length )
```

Sets the value of a register node; manually set whether to verify the node.

See also

spinError

Parameters

<i>hNode</i>	The register node of the value to set
<i>bVerify</i>	The boolean of whether to verify the node
<i>pBuf</i>	The unsigned integer buffer of the value to set
<i>length</i>	The number of bytes of the value to set

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.69 spinRegisterSetReference()

```
SPINNAKERC_API spinRegisterSetReference (
    spinNodeHandle hNode,
    spinNodeHandle hRef )
```

Uses a second node as a reference for a register node.

See also

spinError

Parameters

<i>hNode</i>	The register node that houses the reference
<i>hRef</i>	The reference node

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.70 spinStringGetMaxLength()

```
SPINNAKERC_API spinStringGetMaxLength (
    spinNodeHandle hNode,
    int64_t * pValue )
```

Retrieves the maximum length of the c-string to be returned.

See also

spinError

Parameters

<i>hNode</i>	The string node of the length to retrieve
<i>pValue</i>	The integer pointer in which the maximum length of the c-string is returned

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.71 spinStringGetValue()

```
SPINNAKERC_API spinStringGetValue (
    spinNodeHandle hNode,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the value of a string node as a c-string.

See also

spinError

Parameters

<i>hNode</i>	The string node of the value to read
<i>pBuf</i>	The c-string character buffer in which the value of the node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.72 spinStringGetValueEx()

```
SPINNAKERC_API spinStringGetValueEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    char * pBuf,
    size_t * pBufLen )
```

Retrieves the value of a string node as a cstring; manually set whether to verify the node.

See also

spinError

Parameters

<i>hNode</i>	The string node of the value to read
<i>bVerify</i>	The boolean of whether to verify the node
<i>pBuf</i>	The c-string character buffer in which the value of the node is returned
<i>pBufLen</i>	The unsigned integer pointer in which the length of the c-string is returned; the input value is the maximum length

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.73 spinStringSetValue()

```
SPINNAKERC_API spinStringSetValue (
    spinNodeHandle hNode,
    const char * pBuf )
```

Sets the value of a string node.

See also

spinError

Parameters

<i>hNode</i>	The string node having its value changed
<i>pBuf</i>	The c-string of the value to set

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.14.1.74 spinStringSetValueEx()

```
SPINNAKERC_API spinStringSetValueEx (
    spinNodeHandle hNode,
    bool8_t bVerify,
    const char * pBuf )
```

Sets the value of a string node; manually set whether to verify the node.

See also

spinError

Parameters

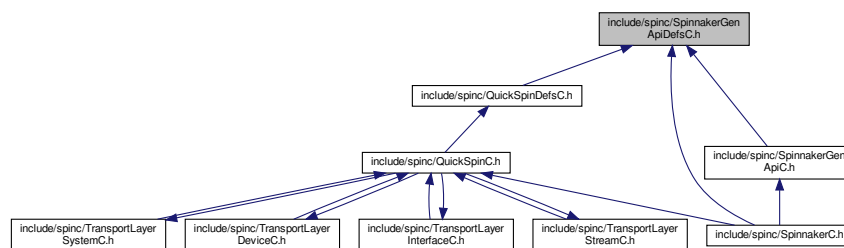
<i>hNode</i>	The string node having its value changed
<i>bVerify</i>	The boolean of whether to verify the node
<i>pBuf</i>	The c-string of the value to set

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.15 include/spinc/SpinnakerGenApiDefsC.h File Reference

This graph shows which files directly or indirectly include this file:



Typedefs

- typedef void * [spinNodeMapHandle](#)
Handle for nodemap functionality.
- typedef void * [spinNodeHandle](#)
Handle for node functionality.
- typedef void * [spinNodeCallbackHandle](#)
Handle for callback functionality.
- typedef void(* [spinNodeCallbackFunction](#)) ([spinNodeHandle](#) hNode)
Function signatures are used to create and trigger callbacks and events.

Enumerations

- enum [_spinNodeType](#) {
[ValueNode](#),
[BaseNode](#),
[IntegerNode](#),
[BooleanNode](#),
[FloatNode](#),
[CommandNode](#),
[StringNode](#),
[RegisterNode](#),
[EnumerationNode](#),
[EnumEntryNode](#),
[CategoryNode](#),
[PortNode](#),
[UnknownNode](#) = -1 }

- enum `_spinSign` {
`Signed`,
`Unsigned`,
`_UndefinedSign` }
- enum `_spinAccessMode` {
`NI`,
`NA`,
`WO`,
`RO`,
`RW`,
`_UndefinedAccesMode`,
`_CycleDetectAccesMode` }
- enum `_spinVisibility` {
`Beginner` = 0,
`Expert` = 1,
`Guru` = 2,
`Invisible` = 3,
`_UndefinedVisibility` = 99 }
- enum `_spinCachingMode` {
`NoCache`,
`WriteThrough`,
`WriteAround`,
`_UndefinedCachingMode` }
- enum `_spinRepresentation` {
`Linear`,
`Logarithmic`,
`Boolean`,
`PureNumber`,
`HexNumber`,
`IPV4Address`,
`MACAddress`,
`_UndefinedRepresentation` }
recommended representation of a node value
- enum `_spinEndianness` {
`BigEndian`,
`LittleEndian`,
`_UndefinedEndian` }
Endianness of a value in a register.
- enum `_spinNameSpace` {
`Custom`,
`Standard`,
`_UndefinedNameSpace` }
Defines if a node name is standard or custom.
- enum `_spinStandardNameSpace` {
`None`,
`GEV`,
`IIDC`,
`CL`,
`USB`,
`_UndefinedStandardNameSpace` }
Defines from which standard namespace a node name comes from.
- enum `_spinYesNo` {
`Yes` = 1,
`No` = 0,
`_UndefinedYesNo` = 2 }
Defines the chices of a Yes/No alternaitve.

- enum `_spinSlope` {
`Increasing`,
`Decreasing`,
`Varying`,
`Automatic`,
`_UndefinedESlope` }

typedef for fomula type

- enum `_spinXMLValidation` {
`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 `_spinDisplayNotation` {
`fnAutomatic`,
`fnFixed`,
`fnScientific`,
`_UndefinedEDisplayNotation` }

typedef for float notation

- enum `_spinInterfaceType` {
`intfIValue`,
`intfIBase`,
`intfInteger`,
`intfBoolean`,
`intfCommand`,
`intfFloat`,
`intfString`,
`intfRegister`,
`intfCategory`,
`intfEnumeration`,
`intfEnumEntry`,
`intfIPort` }

typedef for interface type

- enum `_spinLinkType` {
`ctAllDependingNodes`,
`ctAllTerminalNodes`,
`ctInvalidators`,
`ctReadingChildren`,
`ctWritingChildren`,
`ctDependingChildren` }

typedef for link type

- enum `_spinIncMode` {
`noIncrement`,
`fixedIncrement`,
`listIncrement` }

typedef for increment mode

- enum `_spinInputDirection` {
`idFrom`,
`idTo`,
`idNone` }

typedef for link type

13.15.1 Typedef Documentation

13.15.1.1 spinNodeCallbackFunction

```
typedef void(* spinNodeCallbackFunction) (spinNodeHandle hNode)
```

Function signatures are used to create and trigger callbacks and events.

13.15.1.2 spinNodeCallbackHandle

```
typedef void* spinNodeCallbackHandle
```

Handle for callback functionality.

Created by calling [spinNodeRegisterCallback\(\)](#), which requires a call to [spinNodeUnregisterCallback\(\)](#) destroy.

13.15.1.3 spinNodeHandle

```
typedef void* spinNodeHandle
```

Handle for node functionality.

Created by calling [spinNodeMapGetNode\(\)](#). No need to release, clear, or destroy.

13.15.1.4 spinNodeMapHandle

```
typedef void* spinNodeMapHandle
```

Handle for nodemap functionality.

Created by calling [spinCameraGetNodemap\(\)](#), [spinCameraGetTLDeviceNodeMap\(\)](#), [spinCameraGetTLStreamNodeMap\(\)](#) or [spinInterfaceGetTLNodeMap\(\)](#). No need to release, clear, or destroy.

13.15.2 Enumeration Type Documentation

13.15.2.1 _spinAccessMode

```
enum _spinAccessMode
```

Enumerator

NI	
NA	
WO	
RO	
RW	
_UndefinedAccesMode	
_CycleDetectAccesMode	

13.15.2.2 _spinCachingMode

```
enum _spinCachingMode
```

Enumerator

NoCache	
WriteThrough	
WriteAround	
_UndefinedCachingMode	

13.15.2.3 _spinDisplayNotation

```
enum _spinDisplayNotation
```

```
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

13.15.2.4 `_spinEndianness`

enum `_spinEndianness`

Endianness of a value in a register.

Enumerator

BigEndian	Register is big endian.
LittleEndian	Register is little endian.
_UndefinedEndian	Object is not yet initialized.

13.15.2.5 `_spinIncMode`

enum `_spinIncMode`

typedef for increment mode

Enumerator

noIncrement	
fixedIncrement	
listIncrement	

13.15.2.6 `_spinInputDirection`

enum `_spinInputDirection`

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

13.15.2.7 `_spinInterfaceType`

enum `_spinInterfaceType`

typedef for interface type

Enumerator

<code>intflValue</code>	
<code>intflBase</code>	IValue interface
<code>intflInteger</code>	IBase interface
<code>intflBoolean</code>	IInteger interface
<code>intflCommand</code>	IBoolean interface
<code>intflFloat</code>	ICommand interface
<code>intflString</code>	IFloat interface
<code>intflRegister</code>	IString interface
<code>intflCategory</code>	IRegister interface
<code>intflEnumeration</code>	ICategory interface
<code>intflEnumEntry</code>	IEnumeration interface
<code>intflPort</code>	IEnumEntry interface IPort interface

13.15.2.8 `_spinLinkType`

enum `_spinLinkType`

typedef for link type

Enumerator

ctAllDependingNodes	
ctAllTerminalNodes	All nodes which will be invalidated if this node becomes invalid
ctInvalidators	All terminal nodes which may be written to by this node
ctReadingChildren	List of references to nodes which may invalidate this node
ctWritingChildren	All child nodes which influence this node's AccessMode
ctDependingChildren	All child nodes which may be written to All child nodes which will cause this node to be invalidated

13.15.2.9 _spinNameSpace

```
enum _spinNameSpace
```

Defines if a node name is standard or custom.

Enumerator

Custom	name resides in custom namespace
Standard	name resides in one of the standard namespaces
_UndefinedNameSpace	Object is not yet initialized.

13.15.2.10 _spinNodeType

```
enum _spinNodeType
```

Enumerator

ValueNode	
BaseNode	
IntegerNode	
BooleanNode	
FloatNode	
CommandNode	
StringNode	

Enumerator

RegisterNode	
EnumerationNode	
EnumEntryNode	
CategoryNode	
PortNode	
UnknownNode	

13.15.2.11 _spinRepresentation

```
enum _spinRepresentation
```

recommended representation of a node value

Enumerator

Linear	Slider with linear behavior.
Logarithmic	Slider with logarithmic behaviour.
Boolean	Check box.
PureNumber	Decimal number in an edit control.
HexNumber	Hex number in an edit control.
IPV4Address	IP-Address.
MACAddress	MAC-Address.
_UndefinedRepresentation	

13.15.2.12 _spinSign

```
enum _spinSign
```

Enumerator

Signed	
Unsigned	
_UndefinedSign	

13.15.2.13 _spinSlope

```
enum _spinSlope
```

typedef for fomula 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

13.15.2.14 _spinStandardNameSpace

```
enum _spinStandardNameSpace
```

Defines from which standard namespace a node name comes from.

Enumerator

None	name resides in custom namespace
GEV	name resides in GigE Vision namespace
IIDC	name resides in 1394 IIDC namespace
CL	name resides in camera link namespace
USB	name resides in USB namespace
_UndefinedStandardNameSpace	Object is not yet initialized.

13.15.2.15 _spinVisibility

```
enum _spinVisibility
```

Enumerator

Beginner	
Expert	
Guru	
Invisible	
_UndefinedVisibility	

13.15.2.16 _spinXMLValidation

```
enum _spinXMLValidation
```

typedef describing the different validity checks which can be performed on an XML file

The enum values for a bitfield 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 Object is not yet initialized

13.15.2.17 _spinYesNo

```
enum _spinYesNo
```

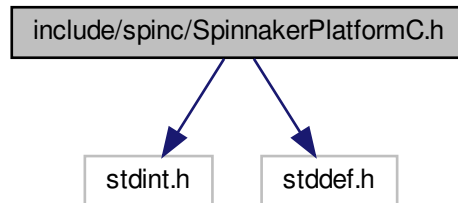
Defines the chices of a Yes/No alternative.

Enumerator

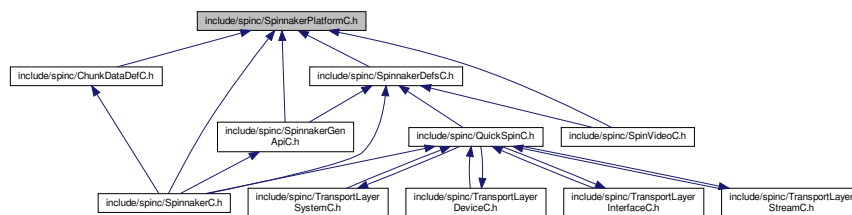
Yes	yes
No	no
_UndefinedYesNo	Object is not yet initialized.

13.16 include/spinc/SpinnakerPlatformC.h File Reference

Include dependency graph for SpinnakerPlatformC.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define SPINNAKERC_API SPINC_IMPORT_EXPORT spinError SPINC_CALLTYPE`

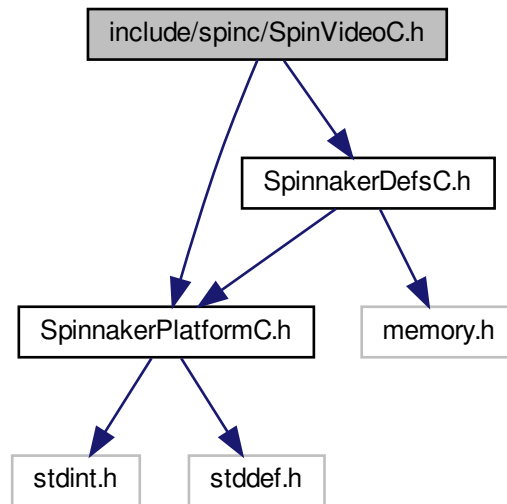
13.16.1 Macro Definition Documentation

13.16.1.1 SPINNAKERC_API

```
#define SPINNAKERC_API SPINC_IMPORT_EXPORT spinError SPINC_CALLTYPE
```

13.17 include/spinc/SpinVideoC.h File Reference

Include dependency graph for SpinVideoC.h:



Functions

- [SPINNAKERC_API spinVideoOpenUncompressed](#) ([spinVideo](#) *phSpinVideo, const char *pName, spinAVIOOption option)
- [SPINNAKERC_API spinVideoOpenMJPEG](#) ([spinVideo](#) *phSpinVideo, const char *pName, spinMJPEGOption option)
- [SPINNAKERC_API spinVideoOpenH264](#) ([spinVideo](#) *phSpinVideo, const char *pName, spinH264Option option)
- [SPINNAKERC_API spinVideoAppend](#) ([spinVideo](#) hSpinVideo, [spinImage](#) hImage)
- [SPINNAKERC_API spinVideoSetMaximumFileSize](#) ([spinVideo](#) hSpinVideo, unsigned int size)
Set the maximum file size (in megabytes) of a AVI/MP4 file.
- [SPINNAKERC_API spinVideoClose](#) ([spinVideo](#) hSpinVideo)

13.17.1 Function Documentation

13.17.1.1 spinVideoAppend()

```

SPINNAKERC_API spinVideoAppend (
    spinVideo hSpinVideo,
    spinImage hImage )

```

13.17.1.2 spinVideoClose()

```
SPINNAKERC_API spinVideoClose (
    spinVideo hSpinVideo )
```

13.17.1.3 spinVideoOpenH264()

```
SPINNAKERC_API spinVideoOpenH264 (
    spinVideo * phSpinVideo,
    const char * pName,
    spinH264Option option )
```

13.17.1.4 spinVideoOpenMJPEG()

```
SPINNAKERC_API spinVideoOpenMJPEG (
    spinVideo * phSpinVideo,
    const char * pName,
    spinMJPEGOption option )
```

13.17.1.5 spinVideoOpenUncompressed()

```
SPINNAKERC_API spinVideoOpenUncompressed (
    spinVideo * phSpinVideo,
    const char * pName,
    spinAVIOption option )
```

13.17.1.6 spinVideoSetMaximumFileSize()

```
SPINNAKERC_API spinVideoSetMaximumFileSize (
    spinVideo hSpinVideo,
    unsigned int size )
```

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

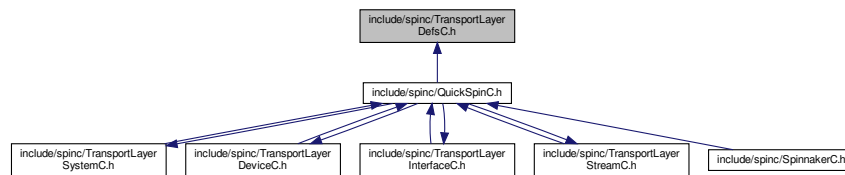
<i>hSpinVideo</i>	The spin video recorder to append the image to
<i>size</i>	The maximum video file size in MB.

Returns

spinError The error code; returns SPINNAKER_ERR_SUCCESS (or 0) for no error

13.18 include/spinc/TransportLayerDefsC.h File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

- enum `_spinTLStreamTypeEnums` {
`StreamType_GigEVision`,
`StreamType_CameraLink`,
`StreamType_CameraLinkHS`,
`StreamType_CoaXPRESS`,
`StreamType_USB3Vision`,
`StreamType_Custom`,
`NUMSTREAMTYPE` }

The enumeration definitions for transport layer nodes.

- enum `_spinTLStreamModeEnums` {
`StreamMode_Socket`,
`StreamMode_LWF`,
`StreamMode_MVA`,
`NUMSTREAMMODE` }
- enum `_spinTLStreamBufferCountModeEnums` {
`StreamBufferCountMode_Manual`,
`NUMSTREAMBUFFERCOUNTMODE` }
- enum `_spinTLStreamBufferHandlingModeEnums` {
`StreamBufferHandlingMode_OldestFirst`,
`StreamBufferHandlingMode_OldestFirstOverwrite`,
`StreamBufferHandlingMode_NewestOnly`,
`StreamBufferHandlingMode_NewestFirst`,
`NUMSTREAMBUFFERHANDLINGMODE` }
- enum `_spinTLDeviceTypeEnums` {
`DeviceType_GigEVision`,
`DeviceType_CameraLink`,
`DeviceType_CameraLinkHS`,
`DeviceType_CoaXPRESS`,
`DeviceType_USB3Vision`,
`DeviceType_Custom`,
`NUMDEVICETYPE` }

- `enum _spinTLDeviceAccessStatusEnums {`
 `DeviceAccessStatus_Unknown,`
 `DeviceAccessStatus_ReadWrite,`
 `DeviceAccessStatus_ReadOnly,`
 `DeviceAccessStatus_NoAccess,`
 `DeviceAccessStatus_Busy,`
 `DeviceAccessStatus_OpenReadWrite,`
 `DeviceAccessStatus_OpenReadOnly,`
 `NUMDEVICEACCESSSTATUS }`
- `enum _spinTLGenICamXMLLocationEnums {`
 `GenICamXMLLocation_Device,`
 `GenICamXMLLocation_Host,`
 `NUMGENICAMXMLLOCATION }`
- `enum _spinTLGUIXMLLocationEnums {`
 `GUIXMLLocation_Device,`
 `GUIXMLLocation_Host,`
 `NUMGUIXMLLOCATION }`
- `enum _spinTLGevCCPEnums {`
 `GevCCP_EnumEntry_GevCCP_OpenAccess,`
 `GevCCP_EnumEntry_GevCCP_ExclusiveAccess,`
 `GevCCP_EnumEntry_GevCCP_ControlAccess,`
 `NUMGEVCCP }`
- `enum _spinTLDeviceEndianessMechanismEnums {`
 `DeviceEndianessMechanism_Legacy,`
 `DeviceEndianessMechanism_Standard,`
 `NUMDEVICEENDIANESSMECHANISM }`
- `enum _spinTLDeviceCurrentSpeedEnums {`
 `DeviceCurrentSpeed_UnknownSpeed,`
 `DeviceCurrentSpeed_LowSpeed,`
 `DeviceCurrentSpeed_FullSpeed,`
 `DeviceCurrentSpeed_HighSpeed,`
 `DeviceCurrentSpeed_SuperSpeed,`
 `NUMDEVICECURRENTSPEED }`
- `enum _spinTLInterfaceTypeEnums {`
 `InterfaceType_GigEVision,`
 `InterfaceType_CameraLink,`
 `InterfaceType_CameraLinkHS,`
 `InterfaceType_CoaxPress,`
 `InterfaceType_USB3Vision,`
 `InterfaceType_Custom,`
 `NUMINTERFACETYPE }`
- `enum _spinTLPOEStatusEnums {`
 `POEStatus_NotSupported,`
 `POEStatus_PowerOff,`
 `POEStatus_PowerOn,`
 `NUMPOESTATUS }`
- `enum _spinTLFilterDriverStatusEnums {`
 `FilterDriverStatus_NotSupported,`
 `FilterDriverStatus_Disabled,`
 `FilterDriverStatus_Enabled,`
 `NUMFILTERDRIVERSTATUS }`
- `enum _spinTLTLTypeEnums {`
 `TLType_GigEVision,`
 `TLType_CameraLink,`
 `TLType_CameraLinkHS,`
 `TLType_CoaxPress,`
 `TLType_USB3Vision,`
 `TLType_Mixed,`

```
TLType_Custom,
NUMTLTYPE }
```

13.18.1 Enumeration Type Documentation

13.18.1.1 _spinTLDeviceAccessStatusEnums

```
enum _spinTLDeviceAccessStatusEnums
```

< Gets the access status the transport layer Producer has on the device.

Enumerator

DeviceAccessStatus_Unknown	Not known to producer.
DeviceAccessStatus_ReadWrite	Full access
DeviceAccessStatus_ReadOnly	Read-only access
DeviceAccessStatus_NoAccess	Not available to connect
DeviceAccessStatus_Busy	The device is already opened by another entity
DeviceAccessStatus_OpenReadWrite	Open in Read/Write mode by this GenTL host
DeviceAccessStatus_OpenReadOnly	Open in Read access mode by this GenTL host
NUMDEVICEACCESSSTATUS	

13.18.1.2 _spinTLDeviceCurrentSpeedEnums

```
enum _spinTLDeviceCurrentSpeedEnums
```

< 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	

13.18.1.3 _spinTLDeviceEndianessMechanismEnums

```
enum _spinTLDeviceEndianessMechanismEnums
```

< 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	

13.18.1.4 _spinTLDeviceTypeEnums

enum [_spinTLDeviceTypeEnums](#)

< Transport layer type of the device.

Enumerator

DeviceType_GigEVision	GigE Vision
DeviceType_CameraLink	Camera Link
DeviceType_CameraLinkHS	Camera Link High Speed
DeviceType_CoaXPress	CoaXPress
DeviceType_USB3Vision	USB3 Vision
DeviceType_Custom	Custom transport layer
NUMDEVICETYPE	

13.18.1.5 _spinTLFilterDriverStatusEnums

enum [_spinTLFilterDriverStatusEnums](#)

< Reports whether FLIR Light Weight Filter Driver is enabled, disabled, or not installed.

Enumerator

FilterDriverStatus_NotSupported	Not Installed
FilterDriverStatus_Disabled	FLIR Light Weight Filter Driver is disabled across all interfaces
FilterDriverStatus_Enabled	FLIR Light Weight Filter Driver is enabled
NUMFILTERDRIVERSTATUS	

13.18.1.6 _spinTLGenICamXMLLocationEnums

enum [_spinTLGenICamXMLLocationEnums](#)

< Sets the location to load GenICam XML.

Enumerator

GenICamXMLLocation_Device	Load GenICam XML from device
GenICamXMLLocation_Host	Load GenICam XML from host
NUMGENICAMXMLLOCATION	

13.18.1.7 _spinTLGevCCPEnums

```
enum _spinTLGevCCPEnums
```

< 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	

13.18.1.8 _spinTLGUIXMLLocationEnums

```
enum _spinTLGUIXMLLocationEnums
```

< Sets the location to load GUI XML.

Enumerator

GUIXMLLocation_Device	Load XML from device
GUIXMLLocation_Host	Load XML from host
NUMGUIXMLLOCATION	

13.18.1.9 _spinTLInterfaceTypeEnums

```
enum _spinTLInterfaceTypeEnums
```

< Transport layer type of the interface.

Enumerator

InterfaceType_GigEVision	GigE Vision
InterfaceType_CameraLink	Camera Link

Enumerator

InterfaceType_CameraLinkHS	Camera Link High Speed
InterfaceType_CoaXPress	CoaXPress
InterfaceType_USB3Vision	USB3 Vision
InterfaceType_Custom	Custom transport layer
NUMINTERFACETYPE	

13.18.1.10 _spinTLPOEStatusEnums

```
enum _spinTLPOEStatusEnums
```

< 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	

13.18.1.11 _spinTLStreamBufferCountModeEnums

```
enum _spinTLStreamBufferCountModeEnums
```

< Controls access to setting the number of buffers used for the stream.

Enumerator

StreamBufferCountMode_Manual	The number of buffers used for the stream is set by the user.
NUMSTREAMBUFFERCOUNTMODE	

13.18.1.12 _spinTLStreamBufferHandlingModeEnums

```
enum _spinTLStreamBufferHandlingModeEnums
```

< 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_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.
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.
NUMSTREAMBUFFERHANDLINGMODE	

13.18.1.13 _spinTLStreamModeEnums

```
enum _spinTLStreamModeEnums
```

< Stream mode of the device.

Enumerator

StreamMode_Socket	Socket
StreamMode_LWF	Light Weight Filter Driver
StreamMode_MVA	Machine Vision Accelerator Driver
NUMSTREAMMODE	

13.18.1.14 _spinTLStreamTypeEnums

```
enum _spinTLStreamTypeEnums
```

The enumeration definitions for transport layer nodes.

< Stream type of the device.

Enumerator

StreamType_GigEVision	GigE Vision
StreamType_CameraLink	Camera Link
StreamType_CameraLinkHS	Camera Link High Speed
StreamType_CoaXPress	CoaXPress
StreamType_USB3Vision	USB3 Vision
StreamType_Custom	Custom transport layer
NUMSTREAMTYPE	

13.18.1.15 _spinTLTLTypeEnums

enum `_spinTLTLTypeEnums`

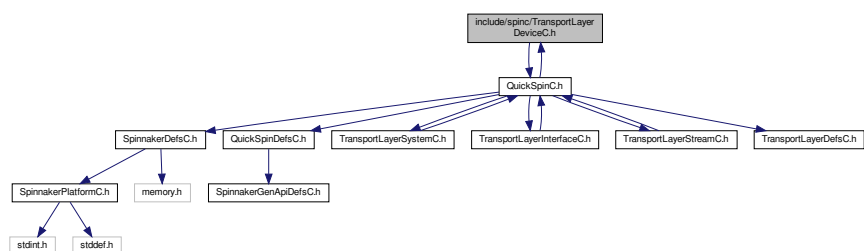
< Transport layer type of the GenTL Producer implementation.

Enumerator

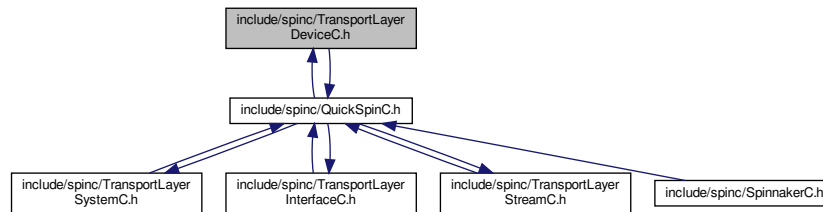
TLType_GigEVision	GigE Vision
TLType_CameraLink	Camera Link
TLType_CameraLinkHS	Camera Link High Speed
TLType_CoaXPress	CoaXPress
TLType_USB3Vision	USB3 Vision
TLType_Mixed	Different Interface modules of the GenTL Producer are of different types
TLType_Custom	Custom transport layer
NUMTLTYPE	

13.19 include/spinc/TransportLayerDeviceC.h File Reference

Include dependency graph for TransportLayerDeviceC.h:



This graph shows which files directly or indirectly include this file:

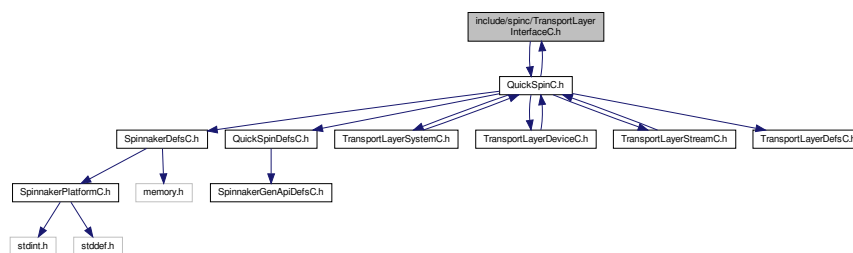


Data Structures

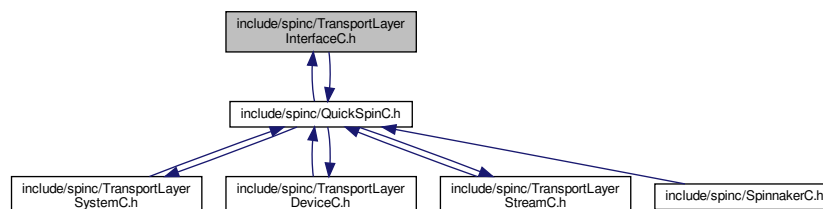
- [struct _quickSpinTLDevice](#)

13.20 include/spinc/TransportLayerInterfaceC.h File Reference

Include dependency graph for TransportLayerInterfaceC.h:



This graph shows which files directly or indirectly include this file:

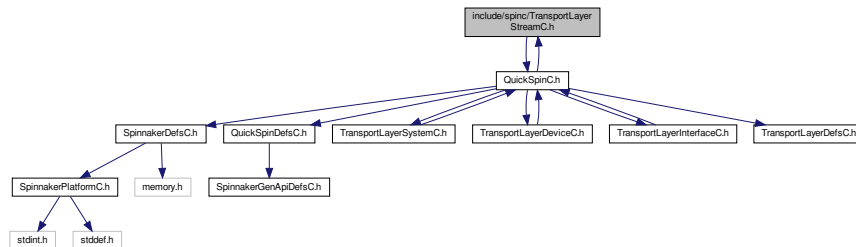


Data Structures

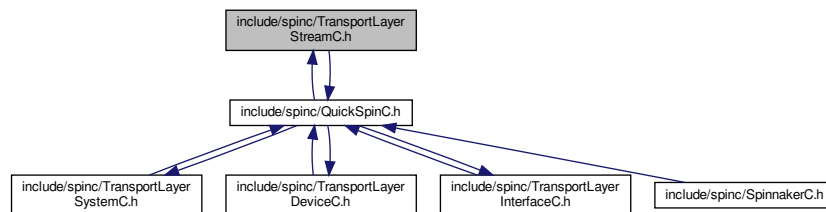
- [struct _quickSpinTLInterface](#)

13.21 include/spinc/TransportLayerStreamC.h File Reference

Include dependency graph for TransportLayerStreamC.h:



This graph shows which files directly or indirectly include this file:

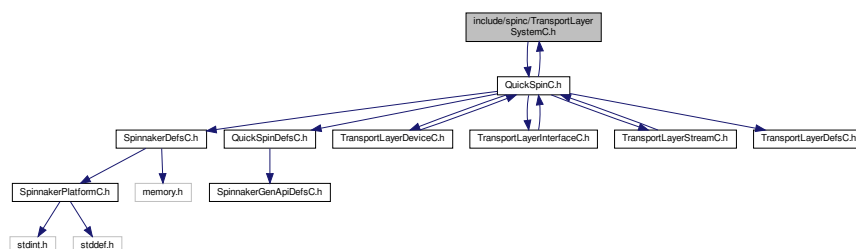


Data Structures

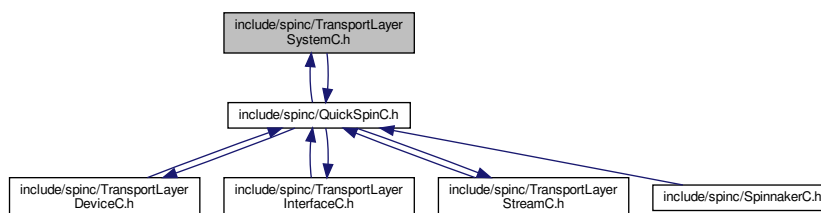
- struct [_quickSpinTLStream](#)

13.22 include/spinc/TransportLayerSystemC.h File Reference

Include dependency graph for TransportLayerSystemC.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- [struct _quickSpinTLSystem](#)

Index

- [_CycleDetectAccessMode](#)
 - [SpinnakerGenApiDefsC.h, 497](#)
- [_UndefinedAccessMode](#)
 - [SpinnakerGenApiDefsC.h, 497](#)
- [_UndefinedCachingMode](#)
 - [SpinnakerGenApiDefsC.h, 497](#)
- [_UndefinedEDisplayNotation](#)
 - [SpinnakerGenApiDefsC.h, 497](#)
- [_UndefinedESlope](#)
 - [SpinnakerGenApiDefsC.h, 502](#)
- [_UndefinedEXMLValidation](#)
 - [SpinnakerGenApiDefsC.h, 503](#)
- [_UndefinedEndian](#)
 - [SpinnakerGenApiDefsC.h, 498](#)
- [_UndefinedNameSpace](#)
 - [SpinnakerGenApiDefsC.h, 500](#)
- [_UndefinedRepresentation](#)
 - [SpinnakerGenApiDefsC.h, 501](#)
- [_UndefinedSign](#)
 - [SpinnakerGenApiDefsC.h, 501](#)
- [_UndefinedStandardNameSpace](#)
 - [SpinnakerGenApiDefsC.h, 502](#)
- [_UndefinedVisibility](#)
 - [SpinnakerGenApiDefsC.h, 502](#)
- [_UndefinedYesNo](#)
 - [SpinnakerGenApiDefsC.h, 503](#)
- [_actionCommandResult, 69](#)
 - [DeviceAddress, 69](#)
 - [Status, 69](#)
- [_quickSpin, 70](#)
 - [AasRoiEnable, 82](#)
 - [AasRoiHeight, 82](#)
 - [AasRoiOffsetX, 82](#)
 - [AasRoiOffsetY, 82](#)
 - [AasRoiWidth, 82](#)
 - [AcquisitionAbort, 83](#)
 - [AcquisitionArm, 83](#)
 - [AcquisitionBurstFrameCount, 83](#)
 - [AcquisitionFrameCount, 83](#)
 - [AcquisitionFrameRate, 83](#)
 - [AcquisitionFrameRateEnable, 83](#)
 - [AcquisitionLineRate, 83](#)
 - [AcquisitionMode, 83](#)
 - [AcquisitionResultingFrameRate, 84](#)
 - [AcquisitionStart, 84](#)
 - [AcquisitionStatus, 84](#)
 - [AcquisitionStatusSelector, 84](#)
 - [AcquisitionStop, 84](#)
 - [ActionDeviceKey, 84](#)
 - [ActionGroupKey, 84](#)
 - [ActionGroupMask, 84](#)
 - [ActionQueueSize, 85](#)
 - [ActionSelector, 85](#)
 - [ActionUnconditionalMode, 85](#)
 - [AdaptiveCompressionEnable, 85](#)
 - [AdcBitDepth, 85](#)
 - [aPAUSEMACCtrlFramesReceived, 85](#)
 - [aPAUSEMACCtrlFramesTransmitted, 85](#)
 - [AutoAlgorithmSelector, 85](#)
 - [AutoExposureControlLoopDamping, 86](#)
 - [AutoExposureControlPriority, 86](#)
 - [AutoExposureEVCompensation, 86](#)
 - [AutoExposureExposureTimeLowerLimit, 86](#)
 - [AutoExposureExposureTimeUpperLimit, 86](#)
 - [AutoExposureGainLowerLimit, 86](#)
 - [AutoExposureGainUpperLimit, 86](#)
 - [AutoExposureGreyValueLowerLimit, 86](#)
 - [AutoExposureGreyValueUpperLimit, 87](#)
 - [AutoExposureLightingMode, 87](#)
 - [AutoExposureMeteringMode, 87](#)
 - [AutoExposureTargetGreyValue, 87](#)
 - [AutoExposureTargetGreyValueAuto, 87](#)
 - [BalanceRatio, 87](#)
 - [BalanceRatioSelector, 87](#)
 - [BalanceWhiteAuto, 87](#)
 - [BalanceWhiteAutoDamping, 88](#)
 - [BalanceWhiteAutoLowerLimit, 88](#)
 - [BalanceWhiteAutoProfile, 88](#)
 - [BalanceWhiteAutoUpperLimit, 88](#)
 - [BinningHorizontal, 88](#)
 - [BinningHorizontalMode, 88](#)
 - [BinningSelector, 88](#)
 - [BinningVertical, 88](#)
 - [BinningVerticalMode, 89](#)
 - [BlackLevel, 89](#)
 - [BlackLevelAuto, 89](#)
 - [BlackLevelAutoBalance, 89](#)
 - [BlackLevelClampingEnable, 89](#)
 - [BlackLevelRaw, 89](#)
 - [BlackLevelSelector, 89](#)
 - [ChunkBlackLevel, 89](#)
 - [ChunkBlackLevelSelector, 90](#)
 - [ChunkCompressionMode, 90](#)
 - [ChunkCompressionRatio, 90](#)
 - [ChunkCounterSelector, 90](#)
 - [ChunkCounterValue, 90](#)
 - [ChunkCRC, 90](#)
 - [ChunkEnable, 90](#)

- ChunkEncoderSelector, 90
- ChunkEncoderStatus, 91
- ChunkEncoderValue, 91
- ChunkExposureEndLineStatusAll, 91
- ChunkExposureTime, 91
- ChunkExposureTimeSelector, 91
- ChunkFrameID, 91
- ChunkGain, 91
- ChunkGainSelector, 91
- ChunkHeight, 92
- ChunkImage, 92
- ChunkImageComponent, 92
- ChunkInferenceBoundingBoxResult, 92
- ChunkInferenceConfidence, 92
- ChunkInferenceFrameID, 92
- ChunkInferenceResult, 92
- ChunkLinePitch, 92
- ChunkLineStatusAll, 93
- ChunkModeActive, 93
- ChunkOffsetX, 93
- ChunkOffsetY, 93
- ChunkPartSelector, 93
- ChunkPixelDynamicRangeMax, 93
- ChunkPixelDynamicRangeMin, 93
- ChunkPixelFormat, 93
- ChunkRegionID, 94
- ChunkScan3dAxisMax, 94
- ChunkScan3dAxisMin, 94
- ChunkScan3dCoordinateOffset, 94
- ChunkScan3dCoordinateReferenceSelector, 94
- ChunkScan3dCoordinateReferenceValue, 94
- ChunkScan3dCoordinateScale, 94
- ChunkScan3dCoordinateSelector, 94
- ChunkScan3dCoordinateSystem, 95
- ChunkScan3dCoordinateSystemReference, 95
- ChunkScan3dCoordinateTransformSelector, 95
- ChunkScan3dDistanceUnit, 95
- ChunkScan3dInvalidDataFlag, 95
- ChunkScan3dInvalidDataValue, 95
- ChunkScan3dOutputMode, 95
- ChunkScan3dTransformValue, 95
- ChunkScanLineSelector, 96
- ChunkSelector, 96
- ChunkSequencerSetActive, 96
- ChunkSerialData, 96
- ChunkSerialDataLength, 96
- ChunkSerialReceiveOverflow, 96
- ChunkSourceID, 96
- ChunkStreamChannelID, 96
- ChunkTimerSelector, 97
- ChunkTimerValue, 97
- ChunkTimestamp, 97
- ChunkTimestampLatchValue, 97
- ChunkTransferBlockID, 97
- ChunkTransferQueueCurrentBlockCount, 97
- ChunkTransferStreamID, 97
- ChunkWidth, 97
- CIConfiguration, 98
- CITimeSlotsCount, 98
- ColorTransformationEnable, 98
- ColorTransformationSelector, 98
- ColorTransformationValue, 98
- ColorTransformationValueSelector, 98
- CompressionRatio, 98
- CompressionSaturationPriority, 98
- CounterDelay, 99
- CounterDuration, 99
- CounterEventActivation, 99
- CounterEventSource, 99
- CounterReset, 99
- CounterResetActivation, 99
- CounterResetSource, 99
- CounterSelector, 99
- CounterStatus, 100
- CounterTriggerActivation, 100
- CounterTriggerSource, 100
- CounterValue, 100
- CounterValueAtReset, 100
- CxpConnectionSelector, 100
- CxpConnectionTestErrorCount, 100
- CxpConnectionTestMode, 100
- CxpConnectionTestPacketCount, 101
- CxpLinkConfiguration, 101
- CxpLinkConfigurationPreferred, 101
- CxpLinkConfigurationStatus, 101
- CxpPoCxpAuto, 101
- CxpPoCxpStatus, 101
- CxpPoCxpTripReset, 101
- CxpPoCxpTurnOff, 101
- DecimationHorizontal, 102
- DecimationHorizontalMode, 102
- DecimationSelector, 102
- DecimationVertical, 102
- DecimationVerticalMode, 102
- DefectCorrectionMode, 102
- DefectCorrectStaticEnable, 102
- DefectTableApply, 102
- DefectTableCoordinateX, 103
- DefectTableCoordinateY, 103
- DefectTableFactoryRestore, 103
- DefectTableIndex, 103
- DefectTablePixelCount, 103
- DefectTableSave, 103
- Deinterlacing, 103
- DeviceCharacterSet, 103
- DeviceClockFrequency, 104
- DeviceClockSelector, 104
- DeviceConnectionSelector, 104
- DeviceConnectionSpeed, 104
- DeviceConnectionStatus, 104
- DeviceEventChannelCount, 104
- DeviceFamilyName, 104
- DeviceFeaturePersistenceEnd, 104
- DeviceFeaturePersistenceStart, 105
- DeviceFirmwareVersion, 105
- DeviceGenCPVersionMajor, 105

- DeviceGenCPVersionMinor, [105](#)
- DeviceID, [105](#)
- DeviceIndicatorMode, [105](#)
- DeviceLinkBandwidthReserve, [105](#)
- DeviceLinkCommandTimeout, [105](#)
- DeviceLinkConnectionCount, [106](#)
- DeviceLinkCurrentThroughput, [106](#)
- DeviceLinkHeartbeatMode, [106](#)
- DeviceLinkHeartbeatTimeout, [106](#)
- DeviceLinkSelector, [106](#)
- DeviceLinkSpeed, [106](#)
- DeviceLinkThroughputLimit, [106](#)
- DeviceLinkThroughputLimitMode, [106](#)
- DeviceManifestEntrySelector, [107](#)
- DeviceManifestPrimaryURL, [107](#)
- DeviceManifestSchemaMajorVersion, [107](#)
- DeviceManifestSchemaMinorVersion, [107](#)
- DeviceManifestSecondaryURL, [107](#)
- DeviceManifestXMLMajorVersion, [107](#)
- DeviceManifestXMLMinorVersion, [107](#)
- DeviceManifestXMLSubMinorVersion, [107](#)
- DeviceManufacturerInfo, [108](#)
- DeviceMaxThroughput, [108](#)
- DeviceModelName, [108](#)
- DevicePowerSupplySelector, [108](#)
- DeviceRegistersCheck, [108](#)
- DeviceRegistersEndianness, [108](#)
- DeviceRegistersStreamingEnd, [108](#)
- DeviceRegistersStreamingStart, [108](#)
- DeviceRegistersValid, [109](#)
- DeviceReset, [109](#)
- DeviceScanType, [109](#)
- DeviceSerialNumber, [109](#)
- DeviceSerialPortBaudRate, [109](#)
- DeviceSerialPortSelector, [109](#)
- DeviceSFNCVersionMajor, [109](#)
- DeviceSFNCVersionMinor, [109](#)
- DeviceSFNCVersionSubMinor, [110](#)
- DeviceStreamChannelCount, [110](#)
- DeviceStreamChannelEndianness, [110](#)
- DeviceStreamChannelLink, [110](#)
- DeviceStreamChannelPacketSize, [110](#)
- DeviceStreamChannelSelector, [110](#)
- DeviceStreamChannelType, [110](#)
- DeviceTapGeometry, [110](#)
- DeviceTemperature, [111](#)
- DeviceTemperatureSelector, [111](#)
- DeviceTLType, [111](#)
- DeviceTLVersionMajor, [111](#)
- DeviceTLVersionMinor, [111](#)
- DeviceTLVersionSubMinor, [111](#)
- DeviceType, [111](#)
- DeviceUptime, [111](#)
- DeviceUserID, [112](#)
- DeviceVendorName, [112](#)
- DeviceVersion, [112](#)
- EncoderDivider, [112](#)
- EncoderMode, [112](#)
- EncoderOutputMode, [112](#)
- EncoderReset, [112](#)
- EncoderResetActivation, [112](#)
- EncoderResetSource, [113](#)
- EncoderSelector, [113](#)
- EncoderSourceA, [113](#)
- EncoderSourceB, [113](#)
- EncoderStatus, [113](#)
- EncoderTimeout, [113](#)
- EncoderValue, [113](#)
- EncoderValueAtReset, [113](#)
- EnumerationCount, [114](#)
- EventAcquisitionEnd, [114](#)
- EventAcquisitionEndFrameID, [114](#)
- EventAcquisitionEndTimestamp, [114](#)
- EventAcquisitionError, [114](#)
- EventAcquisitionErrorFrameID, [114](#)
- EventAcquisitionErrorTimestamp, [114](#)
- EventAcquisitionStart, [114](#)
- EventAcquisitionStartFrameID, [115](#)
- EventAcquisitionStartTimestamp, [115](#)
- EventAcquisitionTransferEnd, [115](#)
- EventAcquisitionTransferEndFrameID, [115](#)
- EventAcquisitionTransferEndTimestamp, [115](#)
- EventAcquisitionTransferStart, [115](#)
- EventAcquisitionTransferStartFrameID, [115](#)
- EventAcquisitionTransferStartTimestamp, [115](#)
- EventAcquisitionTrigger, [116](#)
- EventAcquisitionTriggerFrameID, [116](#)
- EventAcquisitionTriggerTimestamp, [116](#)
- EventActionLate, [116](#)
- EventActionLateFrameID, [116](#)
- EventActionLateTimestamp, [116](#)
- EventCounter0End, [116](#)
- EventCounter0EndFrameID, [116](#)
- EventCounter0EndTimestamp, [117](#)
- EventCounter0Start, [117](#)
- EventCounter0StartFrameID, [117](#)
- EventCounter0StartTimestamp, [117](#)
- EventCounter1End, [117](#)
- EventCounter1EndFrameID, [117](#)
- EventCounter1EndTimestamp, [117](#)
- EventCounter1Start, [117](#)
- EventCounter1StartFrameID, [118](#)
- EventCounter1StartTimestamp, [118](#)
- EventEncoder0Restarted, [118](#)
- EventEncoder0RestartedFrameID, [118](#)
- EventEncoder0RestartedTimestamp, [118](#)
- EventEncoder0Stopped, [118](#)
- EventEncoder0StoppedFrameID, [118](#)
- EventEncoder0StoppedTimestamp, [118](#)
- EventEncoder1Restarted, [119](#)
- EventEncoder1RestartedFrameID, [119](#)
- EventEncoder1RestartedTimestamp, [119](#)
- EventEncoder1Stopped, [119](#)
- EventEncoder1StoppedFrameID, [119](#)
- EventEncoder1StoppedTimestamp, [119](#)
- EventError, [119](#)

- EventErrorCode, 119
- EventErrorFrameID, 120
- EventErrorTimestamp, 120
- EventExposureEnd, 120
- EventExposureEndFrameID, 120
- EventExposureEndTimestamp, 120
- EventExposureStart, 120
- EventExposureStartFrameID, 120
- EventExposureStartTimestamp, 120
- EventFrameBurstEnd, 121
- EventFrameBurstEndFrameID, 121
- EventFrameBurstEndTimestamp, 121
- EventFrameBurstStart, 121
- EventFrameBurstStartFrameID, 121
- EventFrameBurstStartTimestamp, 121
- EventFrameEnd, 121
- EventFrameEndFrameID, 121
- EventFrameEndTimestamp, 122
- EventFrameStart, 122
- EventFrameStartFrameID, 122
- EventFrameStartTimestamp, 122
- EventFrameTransferEnd, 122
- EventFrameTransferEndFrameID, 122
- EventFrameTransferEndTimestamp, 122
- EventFrameTransferStart, 122
- EventFrameTransferStartFrameID, 123
- EventFrameTransferStartTimestamp, 123
- EventFrameTrigger, 123
- EventFrameTriggerFrameID, 123
- EventFrameTriggerTimestamp, 123
- EventLine0AnyEdge, 123
- EventLine0AnyEdgeFrameID, 123
- EventLine0AnyEdgeTimestamp, 123
- EventLine0FallingEdge, 124
- EventLine0FallingEdgeFrameID, 124
- EventLine0FallingEdgeTimestamp, 124
- EventLine0RisingEdge, 124
- EventLine0RisingEdgeFrameID, 124
- EventLine0RisingEdgeTimestamp, 124
- EventLine1AnyEdge, 124
- EventLine1AnyEdgeFrameID, 124
- EventLine1AnyEdgeTimestamp, 125
- EventLine1FallingEdge, 125
- EventLine1FallingEdgeFrameID, 125
- EventLine1FallingEdgeTimestamp, 125
- EventLine1RisingEdge, 125
- EventLine1RisingEdgeFrameID, 125
- EventLine1RisingEdgeTimestamp, 125
- EventLinkSpeedChange, 125
- EventLinkSpeedChangeFrameID, 126
- EventLinkSpeedChangeTimestamp, 126
- EventLinkTrigger0, 126
- EventLinkTrigger0FrameID, 126
- EventLinkTrigger0Timestamp, 126
- EventLinkTrigger1, 126
- EventLinkTrigger1FrameID, 126
- EventLinkTrigger1Timestamp, 126
- EventNotification, 127
- EventSelector, 127
- EventSequencerSetChange, 127
- EventSequencerSetChangeFrameID, 127
- EventSequencerSetChangeTimestamp, 127
- EventSerialData, 127
- EventSerialDataLength, 127
- EventSerialPortReceive, 127
- EventSerialPortReceiveTimestamp, 128
- EventSerialReceiveOverflow, 128
- EventStream0TransferBlockEnd, 128
- EventStream0TransferBlockEndFrameID, 128
- EventStream0TransferBlockEndTimestamp, 128
- EventStream0TransferBlockStart, 128
- EventStream0TransferBlockStartFrameID, 128
- EventStream0TransferBlockStartTimestamp, 128
- EventStream0TransferBlockTrigger, 129
- EventStream0TransferBlockTriggerFrameID, 129
- EventStream0TransferBlockTriggerTimestamp, 129
- EventStream0TransferBurstEnd, 129
- EventStream0TransferBurstEndFrameID, 129
- EventStream0TransferBurstEndTimestamp, 129
- EventStream0TransferBurstStart, 129
- EventStream0TransferBurstStartFrameID, 129
- EventStream0TransferBurstStartTimestamp, 130
- EventStream0TransferEnd, 130
- EventStream0TransferEndFrameID, 130
- EventStream0TransferEndTimestamp, 130
- EventStream0TransferOverflow, 130
- EventStream0TransferOverflowFrameID, 130
- EventStream0TransferOverflowTimestamp, 130
- EventStream0TransferPause, 130
- EventStream0TransferPauseFrameID, 131
- EventStream0TransferPauseTimestamp, 131
- EventStream0TransferResume, 131
- EventStream0TransferResumeFrameID, 131
- EventStream0TransferResumeTimestamp, 131
- EventStream0TransferStart, 131
- EventStream0TransferStartFrameID, 131
- EventStream0TransferStartTimestamp, 131
- EventTest, 132
- EventTestTimestamp, 132
- EventTimer0End, 132
- EventTimer0EndFrameID, 132
- EventTimer0EndTimestamp, 132
- EventTimer0Start, 132
- EventTimer0StartFrameID, 132
- EventTimer0StartTimestamp, 132
- EventTimer1End, 133
- EventTimer1EndFrameID, 133
- EventTimer1EndTimestamp, 133
- EventTimer1Start, 133
- EventTimer1StartFrameID, 133
- EventTimer1StartTimestamp, 133
- ExposureActiveMode, 133
- ExposureAuto, 133
- ExposureMode, 134
- ExposureTime, 134
- ExposureTimeMode, 134

ExposureTimeSelector, [134](#)
FactoryReset, [134](#)
FileAccessBuffer, [134](#)
FileAccessLength, [134](#)
FileAccessOffset, [134](#)
FileOpenMode, [135](#)
FileOperationExecute, [135](#)
FileOperationResult, [135](#)
FileOperationSelector, [135](#)
FileOperationStatus, [135](#)
FileSelector, [135](#)
FileSize, [135](#)
Gain, [135](#)
GainAuto, [136](#)
GainAutoBalance, [136](#)
GainSelector, [136](#)
Gamma, [136](#)
GammaEnable, [136](#)
GevActiveLinkCount, [136](#)
GevCCP, [136](#)
GevCurrentDefaultGateway, [136](#)
GevCurrentIPAddress, [137](#)
GevCurrentIPConfigurationDHCP, [137](#)
GevCurrentIPConfigurationLLA, [137](#)
GevCurrentIPConfigurationPersistentIP, [137](#)
GevCurrentPhysicalLinkConfiguration, [137](#)
GevCurrentSubnetMask, [137](#)
GevDiscoveryAckDelay, [137](#)
GevFirstURL, [137](#)
GevGVCPExtendedStatusCodes, [138](#)
GevGVCPExtendedStatusCodesSelector, [138](#)
GevGVCPHeartbeatDisable, [138](#)
GevGVCPPendingAck, [138](#)
GevGVCPPendingTimeout, [138](#)
GevGVSPExtendedIDMode, [138](#)
GevHeartbeatTimeout, [138](#)
GevIEEE1588, [138](#)
GevIEEE1588ClockAccuracy, [139](#)
GevIEEE1588Mode, [139](#)
GevIEEE1588Status, [139](#)
GevInterfaceSelector, [139](#)
GevIPConfigurationStatus, [139](#)
GevMACAddress, [139](#)
GevMCDA, [139](#)
GevMCPHostPort, [139](#)
GevMCRC, [140](#)
GevMCSP, [140](#)
GevMCTT, [140](#)
GevNumberOfInterfaces, [140](#)
GevPAUSEFrameReception, [140](#)
GevPAUSEFrameTransmission, [140](#)
GevPersistentDefaultGateway, [140](#)
GevPersistentIPAddress, [140](#)
GevPersistentSubnetMask, [141](#)
GevPhysicalLinkConfiguration, [141](#)
GevPrimaryApplicationIPAddress, [141](#)
GevPrimaryApplicationSocket, [141](#)
GevPrimaryApplicationSwitchoverKey, [141](#)
GevSCCFGAllInTransmission, [141](#)
GevSCCFGExtendedChunkData, [141](#)
GevSCCFGPacketResendDestination, [141](#)
GevSCCFGUnconditionalStreaming, [142](#)
GevSCDA, [142](#)
GevSCPD, [142](#)
GevSCPDDirection, [142](#)
GevSCPHostPort, [142](#)
GevSCPInterfaceIndex, [142](#)
GevSCPSBigEndian, [142](#)
GevSCPSDoNotFragment, [142](#)
GevSCPSFireTestPacket, [143](#)
GevSCPSPacketSize, [143](#)
GevSCSP, [143](#)
GevSCZoneConfigurationLock, [143](#)
GevSCZoneCount, [143](#)
GevSCZoneDirectionAll, [143](#)
GevSecondURL, [143](#)
GevStreamChannelSelector, [143](#)
GevSupportedOption, [144](#)
GevSupportedOptionSelector, [144](#)
GevTimestampTickFrequency, [144](#)
GuiXmlManifestAddress, [144](#)
Height, [144](#)
HeightMax, [144](#)
ImageComponentEnable, [144](#)
ImageComponentSelector, [144](#)
ImageCompressionBitrate, [145](#)
ImageCompressionJPEGFormatOption, [145](#)
ImageCompressionMode, [145](#)
ImageCompressionQuality, [145](#)
ImageCompressionRateOption, [145](#)
IspEnable, [145](#)
LineFilterWidth, [145](#)
LineFormat, [145](#)
LineInputFilterSelector, [146](#)
LineInverter, [146](#)
LineMode, [146](#)
LinePitch, [146](#)
LineSelector, [146](#)
LineSource, [146](#)
LineStatus, [146](#)
LineStatusAll, [146](#)
LinkErrorCount, [147](#)
LinkUptime, [147](#)
LogicBlockLUTInputActivation, [147](#)
LogicBlockLUTInputSelector, [147](#)
LogicBlockLUTInputSource, [147](#)
LogicBlockLUTOutputValue, [147](#)
LogicBlockLUTOutputValueAll, [147](#)
LogicBlockLUTRowIndex, [147](#)
LogicBlockLUTSelector, [148](#)
LogicBlockSelector, [148](#)
LUTEnable, [148](#)
LUTIndex, [148](#)
LUTSelector, [148](#)
LUTValue, [148](#)
LUTValueAll, [148](#)

MaxDeviceResetTime, [148](#)
OffsetX, [149](#)
OffsetY, [149](#)
PacketResendRequestCount, [149](#)
PayloadSize, [149](#)
PixelColorFilter, [149](#)
PixelDynamicRangeMax, [149](#)
PixelDynamicRangeMin, [149](#)
PixelFormat, [149](#)
PixelFormatInfoID, [150](#)
PixelFormatInfoSelector, [150](#)
PixelSize, [150](#)
PowerSupplyCurrent, [150](#)
PowerSupplyVoltage, [150](#)
RegionDestination, [150](#)
RegionMode, [150](#)
RegionSelector, [150](#)
ReverseX, [151](#)
ReverseY, [151](#)
RgbTransformLightSource, [151](#)
Saturation, [151](#)
SaturationEnable, [151](#)
Scan3dAxisMax, [151](#)
Scan3dAxisMin, [151](#)
Scan3dCoordinateOffset, [151](#)
Scan3dCoordinateReferenceSelector, [152](#)
Scan3dCoordinateReferenceValue, [152](#)
Scan3dCoordinateScale, [152](#)
Scan3dCoordinateSelector, [152](#)
Scan3dCoordinateSystem, [152](#)
Scan3dCoordinateSystemReference, [152](#)
Scan3dCoordinateTransformSelector, [152](#)
Scan3dDistanceUnit, [152](#)
Scan3dInvalidDataFlag, [153](#)
Scan3dInvalidDataValue, [153](#)
Scan3dOutputMode, [153](#)
Scan3dTransformValue, [153](#)
SensorDescription, [153](#)
SensorDigitizationTaps, [153](#)
SensorHeight, [153](#)
SensorShutterMode, [153](#)
SensorTaps, [154](#)
SensorWidth, [154](#)
SequencerConfigurationMode, [154](#)
SequencerConfigurationValid, [154](#)
SequencerFeatureEnable, [154](#)
SequencerMode, [154](#)
SequencerPathSelector, [154](#)
SequencerSetActive, [154](#)
SequencerSetLoad, [155](#)
SequencerSetNext, [155](#)
SequencerSetSave, [155](#)
SequencerSetSelector, [155](#)
SequencerSetStart, [155](#)
SequencerSetValid, [155](#)
SequencerTriggerActivation, [155](#)
SequencerTriggerSource, [155](#)
SerialPortBaudRate, [156](#)
SerialPortDataBits, [156](#)
SerialPortParity, [156](#)
SerialPortSelector, [156](#)
SerialPortSource, [156](#)
SerialPortStopBits, [156](#)
SerialReceiveFramingErrorCount, [156](#)
SerialReceiveParityErrorCount, [156](#)
SerialReceiveQueueClear, [157](#)
SerialReceiveQueueCurrentCharacterCount, [157](#)
SerialReceiveQueueMaxCharacterCount, [157](#)
SerialTransmitQueueCurrentCharacterCount, [157](#)
SerialTransmitQueueMaxCharacterCount, [157](#)
Sharpening, [157](#)
SharpeningAuto, [157](#)
SharpeningEnable, [157](#)
SharpeningThreshold, [158](#)
SoftwareSignalPulse, [158](#)
SoftwareSignalSelector, [158](#)
SourceCount, [158](#)
SourceSelector, [158](#)
Test0001, [158](#)
TestEventGenerate, [158](#)
TestPattern, [158](#)
TestPatternGeneratorSelector, [159](#)
TestPendingAck, [159](#)
TimerDelay, [159](#)
TimerDuration, [159](#)
TimerReset, [159](#)
TimerSelector, [159](#)
TimerStatus, [159](#)
TimerTriggerActivation, [159](#)
TimerTriggerSource, [160](#)
TimerValue, [160](#)
Timestamp, [160](#)
TimestampLatch, [160](#)
TimestampLatchValue, [160](#)
TimestampReset, [160](#)
TLParamsLocked, [160](#)
TransferAbort, [160](#)
TransferBlockCount, [161](#)
TransferBurstCount, [161](#)
TransferComponentSelector, [161](#)
TransferControlMode, [161](#)
TransferOperationMode, [161](#)
TransferPause, [161](#)
TransferQueueCurrentBlockCount, [161](#)
TransferQueueMaxBlockCount, [161](#)
TransferQueueMode, [162](#)
TransferQueueOverflowCount, [162](#)
TransferResume, [162](#)
TransferSelector, [162](#)
TransferStart, [162](#)
TransferStatus, [162](#)
TransferStatusSelector, [162](#)
TransferStop, [162](#)
TransferStreamChannel, [163](#)
TransferTriggerActivation, [163](#)
TransferTriggerMode, [163](#)

- TransferTriggerSelector, [163](#)
- TransferTriggerSource, [163](#)
- TriggerActivation, [163](#)
- TriggerDelay, [163](#)
- TriggerDivider, [163](#)
- TriggerEventTest, [164](#)
- TriggerMode, [164](#)
- TriggerMultiplier, [164](#)
- TriggerOverlap, [164](#)
- TriggerSelector, [164](#)
- TriggerSoftware, [164](#)
- TriggerSource, [164](#)
- UserOutputSelector, [164](#)
- UserOutputValue, [165](#)
- UserOutputValueAll, [165](#)
- UserOutputValueAllMask, [165](#)
- UserSetDefault, [165](#)
- UserSetFeatureEnable, [165](#)
- UserSetLoad, [165](#)
- UserSetSave, [165](#)
- UserSetSelector, [165](#)
- V3_3Enable, [166](#)
- WhiteClip, [166](#)
- WhiteClipSelector, [166](#)
- Width, [166](#)
- WidthMax, [166](#)
- _quickSpinTLDevice, [167](#)
 - DeviceAccessStatus, [167](#)
 - DeviceBootloaderVersion, [168](#)
 - DeviceCurrentSpeed, [168](#)
 - DeviceDisplayName, [168](#)
 - DeviceDriverVersion, [168](#)
 - DeviceEndianessMechanism, [168](#)
 - DeviceID, [168](#)
 - DeviceInstanceId, [168](#)
 - DevicesUpdater, [168](#)
 - DeviceLinkSpeed, [169](#)
 - DeviceLocation, [169](#)
 - DeviceModelName, [169](#)
 - DeviceMulticastMonitorMode, [169](#)
 - DevicePortId, [169](#)
 - DeviceReset, [169](#)
 - DeviceSerialNumber, [169](#)
 - DeviceType, [169](#)
 - DeviceU3VProtocol, [170](#)
 - DeviceUserID, [170](#)
 - DeviceVendorName, [170](#)
 - DeviceVersion, [170](#)
 - GenICamXMLLocation, [170](#)
 - GenICamXMLPath, [170](#)
 - GevCCP, [170](#)
 - GevDeviceAutoForceIP, [170](#)
 - GevDeviceDiscoverMaximumPacketSize, [171](#)
 - GevDeviceForceGateway, [171](#)
 - GevDeviceForceIP, [171](#)
 - GevDeviceForceIPAddress, [171](#)
 - GevDeviceForceSubnetMask, [171](#)
 - GevDeviceGateway, [171](#)
 - GevDeviceIPAddress, [171](#)
 - GevDevicesWrongSubnet, [171](#)
 - GevDeviceMACAddress, [172](#)
 - GevDeviceMaximumPacketSize, [172](#)
 - GevDeviceMaximumRetryCount, [172](#)
 - GevDeviceModelsBigEndian, [172](#)
 - GevDevicePort, [172](#)
 - GevDeviceReadAndWriteTimeout, [172](#)
 - GevDeviceSubnetMask, [172](#)
 - GevVersionMajor, [172](#)
 - GevVersionMinor, [173](#)
 - GUIXMLLocation, [173](#)
 - GUIXMLPath, [173](#)
- _quickSpinTLInterface, [173](#)
 - ActionCommand, [174](#)
 - DeviceAccessStatus, [174](#)
 - DeviceCount, [174](#)
 - DeviceID, [175](#)
 - DeviceModelName, [175](#)
 - DeviceSelector, [175](#)
 - DeviceSerialNumber, [175](#)
 - DeviceUnlock, [175](#)
 - DeviceUpdateList, [175](#)
 - DeviceVendorName, [175](#)
 - FilterDriverStatus, [175](#)
 - GevActionDeviceKey, [176](#)
 - GevActionGroupKey, [176](#)
 - GevActionGroupMask, [176](#)
 - GevActionTime, [176](#)
 - GevDeviceAutoForceIP, [176](#)
 - GevDeviceForceGateway, [176](#)
 - GevDeviceForceIP, [176](#)
 - GevDeviceForceIPAddress, [176](#)
 - GevDeviceForceSubnetMask, [177](#)
 - GevDeviceGateway, [177](#)
 - GevDeviceIPAddress, [177](#)
 - GevDeviceMACAddress, [177](#)
 - GevDeviceSubnetMask, [177](#)
 - GevInterfaceGateway, [177](#)
 - GevInterfaceGatewaySelector, [177](#)
 - GevInterfaceMACAddress, [177](#)
 - GevInterfaceMTU, [178](#)
 - GevInterfaceReceiveLinkSpeed, [178](#)
 - GevInterfaceSubnetIPAddress, [178](#)
 - GevInterfaceSubnetMask, [178](#)
 - GevInterfaceSubnetSelector, [178](#)
 - GevInterfaceTransmitLinkSpeed, [178](#)
 - HostAdapterDriverVersion, [178](#)
 - HostAdapterName, [178](#)
 - HostAdapterVendor, [179](#)
 - IncompatibleDeviceCount, [179](#)
 - IncompatibleDeviceID, [179](#)
 - IncompatibleDeviceModelName, [179](#)
 - IncompatibleDeviceSelector, [179](#)
 - IncompatibleDeviceVendorName, [179](#)
 - IncompatibleGevDeviceIPAddress, [179](#)
 - IncompatibleGevDeviceMACAddress, [179](#)
 - IncompatibleGevDeviceSubnetMask, [180](#)

- InterfaceDisplayName, [180](#)
- InterfaceID, [180](#)
- InterfaceType, [180](#)
- POEStatus, [180](#)
- _quickSpinTLStream, [181](#)
 - StreamAnnounceBufferMinimum, [181](#)
 - StreamAnnouncedBufferCount, [181](#)
 - StreamBlockTransferSize, [181](#)
 - StreamBufferAlignment, [182](#)
 - StreamBufferCountManual, [182](#)
 - StreamBufferCountMax, [182](#)
 - StreamBufferCountMode, [182](#)
 - StreamBufferCountResult, [182](#)
 - StreamBufferHandlingMode, [182](#)
 - StreamChunkCountMaximum, [182](#)
 - StreamCRCCheckEnable, [182](#)
 - StreamDeliveredFrameCount, [183](#)
 - StreamDroppedFrameCount, [183](#)
 - StreamID, [183](#)
 - StreamIncompleteFrameCount, [183](#)
 - StreamInputBufferCount, [183](#)
 - StreamIsGrabbing, [183](#)
 - StreamLostFrameCount, [183](#)
 - StreamMissedPacketCount, [183](#)
 - StreamMode, [184](#)
 - StreamOutputBufferCount, [184](#)
 - StreamPacketResendEnable, [184](#)
 - StreamPacketResendMaxRequests, [184](#)
 - StreamPacketResendReceivedPacketCount, [184](#)
 - StreamPacketResendRequestCount, [184](#)
 - StreamPacketResendRequestedPacketCount, [184](#)
 - StreamPacketResendRequestSuccessCount, [184](#)
 - StreamPacketResendTimeout, [185](#)
 - StreamReceivedFrameCount, [185](#)
 - StreamReceivedPacketCount, [185](#)
 - StreamStartedFrameCount, [185](#)
 - StreamType, [185](#)
- _quickSpinTLSystem, [186](#)
 - EnumerateGen2Cameras, [186](#)
 - EnumerateGEVInterfaces, [186](#)
 - EnumerateUSBInterfaces, [186](#)
 - GenTLFNCVersionMajor, [187](#)
 - GenTLFNCVersionMinor, [187](#)
 - GenTLFNCVersionSubMinor, [187](#)
 - GenTLVersionMajor, [187](#)
 - GenTLVersionMinor, [187](#)
 - GevInterfaceDefaultGateway, [187](#)
 - GevInterfaceDefaultIPAddress, [187](#)
 - GevInterfaceDefaultSubnetMask, [187](#)
 - GevInterfaceMACAddress, [188](#)
 - GevVersionMajor, [188](#)
 - GevVersionMinor, [188](#)
 - InterfaceDisplayName, [188](#)
 - InterfaceID, [188](#)
 - InterfaceSelector, [188](#)
 - InterfaceUpdateList, [188](#)
 - TLDisplayName, [188](#)
 - TLFileName, [189](#)
 - TLID, [189](#)
 - TLModelName, [189](#)
 - TLPath, [189](#)
 - TLType, [189](#)
 - TLVendorName, [189](#)
 - TLVersion, [189](#)
- _spinAVIOption, [190](#)
 - frameRate, [190](#)
 - height, [190](#)
 - reserved, [190](#)
 - width, [191](#)
- _spinAccessMode
 - SpinnakerGenApiDefsC.h, [496](#)
- _spinAcquisitionModeEnums
 - CameraDefsC.h, [242](#)
- _spinAcquisitionStatusSelectorEnums
 - CameraDefsC.h, [242](#)
- _spinActionCommandStatus
 - SpinnakerDefsC.h, [445](#)
- _spinActionUnconditionalModeEnums
 - CameraDefsC.h, [242](#)
- _spinAdcBitDepthEnums
 - CameraDefsC.h, [243](#)
- _spinAutoAlgorithmSelectorEnums
 - CameraDefsC.h, [243](#)
- _spinAutoExposureControlPriorityEnums
 - CameraDefsC.h, [243](#)
- _spinAutoExposureLightingModeEnums
 - CameraDefsC.h, [244](#)
- _spinAutoExposureMeteringModeEnums
 - CameraDefsC.h, [244](#)
- _spinAutoExposureTargetGreyValueAutoEnums
 - CameraDefsC.h, [245](#)
- _spinBMPOption, [191](#)
 - indexedColor_8bit, [191](#)
 - reserved, [191](#)
- _spinBalanceRatioSelectorEnums
 - CameraDefsC.h, [245](#)
- _spinBalanceWhiteAutoEnums
 - CameraDefsC.h, [245](#)
- _spinBalanceWhiteAutoProfileEnums
 - CameraDefsC.h, [246](#)
- _spinBinningHorizontalModeEnums
 - CameraDefsC.h, [246](#)
- _spinBinningSelectorEnums
 - CameraDefsC.h, [246](#)
- _spinBinningVerticalModeEnums
 - CameraDefsC.h, [247](#)
- _spinBlackLevelAutoBalanceEnums
 - CameraDefsC.h, [247](#)
- _spinBlackLevelAutoEnums
 - CameraDefsC.h, [247](#)
- _spinBlackLevelSelectorEnums
 - CameraDefsC.h, [248](#)
- _spinCachingMode
 - SpinnakerGenApiDefsC.h, [497](#)
- _spinChunkBlackLevelSelectorEnums
 - CameraDefsC.h, [248](#)

- `_spinChunkCounterSelectorEnums`
CameraDefsC.h, [248](#)
- `_spinChunkData`, [192](#)
 - `m_blackLevel`, [193](#)
 - `m_compressionMode`, [193](#)
 - `m_compressionRatio`, [193](#)
 - `m_counterValue`, [193](#)
 - `m_cRC`, [193](#)
 - `m_encoderValue`, [193](#)
 - `m_exposureEndLineStatusAll`, [193](#)
 - `m_exposureTime`, [194](#)
 - `m_frameID`, [194](#)
 - `m_gain`, [194](#)
 - `m_height`, [194](#)
 - `m_image`, [194](#)
 - `m_inferenceConfidence`, [194](#)
 - `m_inferenceFrameID`, [194](#)
 - `m_inferenceResult`, [194](#)
 - `m_linePitch`, [195](#)
 - `m_lineStatusAll`, [195](#)
 - `m_offsetX`, [195](#)
 - `m_offsetY`, [195](#)
 - `m_partSelector`, [195](#)
 - `m_pixelDynamicRangeMax`, [195](#)
 - `m_pixelDynamicRangeMin`, [195](#)
 - `m_scan3dAxisMax`, [195](#)
 - `m_scan3dAxisMin`, [196](#)
 - `m_scan3dCoordinateOffset`, [196](#)
 - `m_scan3dCoordinateReferenceValue`, [196](#)
 - `m_scan3dCoordinateScale`, [196](#)
 - `m_scan3dInvalidDataValue`, [196](#)
 - `m_scan3dTransformValue`, [196](#)
 - `m_scanLineSelector`, [196](#)
 - `m_sequencerSetActive`, [196](#)
 - `m_serialDataLength`, [197](#)
 - `m_streamChannelID`, [197](#)
 - `m_timerValue`, [197](#)
 - `m_timestamp`, [197](#)
 - `m_timestampLatchValue`, [197](#)
 - `m_transferBlockID`, [197](#)
 - `m_transferQueueCurrentBlockCount`, [197](#)
 - `m_width`, [197](#)
- `_spinChunkEncoderSelectorEnums`
CameraDefsC.h, [249](#)
- `_spinChunkEncoderStatusEnums`
CameraDefsC.h, [249](#)
- `_spinChunkExposureTimeSelectorEnums`
CameraDefsC.h, [249](#)
- `_spinChunkGainSelectorEnums`
CameraDefsC.h, [250](#)
- `_spinChunkImageComponentEnums`
CameraDefsC.h, [250](#)
- `_spinChunkPixelFormatEnums`
CameraDefsC.h, [251](#)
- `_spinChunkRegionIDEnums`
CameraDefsC.h, [251](#)
- `_spinChunkScan3dCoordinateReferenceSelectorEnums`
CameraDefsC.h, [251](#)
- `_spinChunkScan3dCoordinateSelectorEnums`
CameraDefsC.h, [252](#)
- `_spinChunkScan3dCoordinateSystemEnums`
CameraDefsC.h, [252](#)
- `_spinChunkScan3dCoordinateSystemReferenceEnums`
CameraDefsC.h, [252](#)
- `_spinChunkScan3dCoordinateTransformSelectorEnums`
CameraDefsC.h, [253](#)
- `_spinChunkScan3dDistanceUnitEnums`
CameraDefsC.h, [253](#)
- `_spinChunkScan3dOutputModeEnums`
CameraDefsC.h, [253](#)
- `_spinChunkSelectorEnums`
CameraDefsC.h, [254](#)
- `_spinChunkSourceIDEnums`
CameraDefsC.h, [255](#)
- `_spinChunkTimerSelectorEnums`
CameraDefsC.h, [255](#)
- `_spinChunkTransferStreamIDEnums`
CameraDefsC.h, [256](#)
- `_spinCIConfigurationEnums`
CameraDefsC.h, [256](#)
- `_spinCITimeSlotsCountEnums`
CameraDefsC.h, [256](#)
- `_spinColorProcessingAlgorithm`
SpinnakerDefsC.h, [446](#)
- `_spinColorTransformationSelectorEnums`
CameraDefsC.h, [257](#)
- `_spinColorTransformationValueSelectorEnums`
CameraDefsC.h, [257](#)
- `_spinCompressionSaturationPriorityEnums`
CameraDefsC.h, [258](#)
- `_spinCounterEventActivationEnums`
CameraDefsC.h, [258](#)
- `_spinCounterEventSourceEnums`
CameraDefsC.h, [258](#)
- `_spinCounterResetActivationEnums`
CameraDefsC.h, [259](#)
- `_spinCounterResetSourceEnums`
CameraDefsC.h, [259](#)
- `_spinCounterSelectorEnums`
CameraDefsC.h, [260](#)
- `_spinCounterStatusEnums`
CameraDefsC.h, [260](#)
- `_spinCounterTriggerActivationEnums`
CameraDefsC.h, [261](#)
- `_spinCounterTriggerSourceEnums`
CameraDefsC.h, [261](#)
- `_spinCxpConnectionTestModeEnums`
CameraDefsC.h, [262](#)
- `_spinCxpLinkConfigurationEnums`
CameraDefsC.h, [262](#)
- `_spinCxpLinkConfigurationPreferredEnums`
CameraDefsC.h, [263](#)
- `_spinCxpLinkConfigurationStatusEnums`
CameraDefsC.h, [264](#)
- `_spinCxpPoCxpStatusEnums`
CameraDefsC.h, [265](#)

- [_spinDecimationHorizontalModeEnums](#)
CameraDefsC.h, [265](#)
- [_spinDecimationSelectorEnums](#)
CameraDefsC.h, [265](#)
- [_spinDecimationVerticalModeEnums](#)
CameraDefsC.h, [266](#)
- [_spinDefectCorrectionModeEnums](#)
CameraDefsC.h, [266](#)
- [_spinDeinterlacingEnums](#)
CameraDefsC.h, [266](#)
- [_spinDeviceCharacterSetEnums](#)
CameraDefsC.h, [267](#)
- [_spinDeviceClockSelectorEnums](#)
CameraDefsC.h, [267](#)
- [_spinDeviceConnectionStatusEnums](#)
CameraDefsC.h, [267](#)
- [_spinDeviceIndicatorModeEnums](#)
CameraDefsC.h, [268](#)
- [_spinDeviceLinkHeartbeatModeEnums](#)
CameraDefsC.h, [268](#)
- [_spinDeviceLinkThroughputLimitModeEnums](#)
CameraDefsC.h, [268](#)
- [_spinDevicePowerSupplySelectorEnums](#)
CameraDefsC.h, [268](#)
- [_spinDeviceRegistersEndiannessEnums](#)
CameraDefsC.h, [269](#)
- [_spinDeviceScanTypeEnums](#)
CameraDefsC.h, [269](#)
- [_spinDeviceSerialPortBaudRateEnums](#)
CameraDefsC.h, [269](#)
- [_spinDeviceSerialPortSelectorEnums](#)
CameraDefsC.h, [270](#)
- [_spinDeviceStreamChannelEndiannessEnums](#)
CameraDefsC.h, [270](#)
- [_spinDeviceStreamChannelTypeEnums](#)
CameraDefsC.h, [270](#)
- [_spinDeviceTLTypeEnums](#)
CameraDefsC.h, [272](#)
- [_spinDeviceTapGeometryEnums](#)
CameraDefsC.h, [271](#)
- [_spinDeviceTemperatureSelectorEnums](#)
CameraDefsC.h, [272](#)
- [_spinDeviceTypeEnums](#)
CameraDefsC.h, [273](#)
- [_spinDisplayNotation](#)
SpinnakerGenApiDefsC.h, [497](#)
- [_spinEncoderModeEnums](#)
CameraDefsC.h, [273](#)
- [_spinEncoderOutputModeEnums](#)
CameraDefsC.h, [273](#)
- [_spinEncoderResetActivationEnums](#)
CameraDefsC.h, [274](#)
- [_spinEncoderResetSourceEnums](#)
CameraDefsC.h, [274](#)
- [_spinEncoderSelectorEnums](#)
CameraDefsC.h, [275](#)
- [_spinEncoderSourceAEnums](#)
CameraDefsC.h, [276](#)
- [_spinEncoderSourceBEnums](#)
CameraDefsC.h, [276](#)
- [_spinEncoderStatusEnums](#)
CameraDefsC.h, [276](#)
- [_spinEndianess](#)
SpinnakerGenApiDefsC.h, [497](#)
- [_spinError](#)
SpinnakerDefsC.h, [446](#)
- [_spinEventNotificationEnums](#)
CameraDefsC.h, [277](#)
- [_spinEventSelectorEnums](#)
CameraDefsC.h, [277](#)
- [_spinExposureActiveModeEnums](#)
CameraDefsC.h, [277](#)
- [_spinExposureAutoEnums](#)
CameraDefsC.h, [277](#)
- [_spinExposureModeEnums](#)
CameraDefsC.h, [278](#)
- [_spinExposureTimeModeEnums](#)
CameraDefsC.h, [278](#)
- [_spinExposureTimeSelectorEnums](#)
CameraDefsC.h, [279](#)
- [_spinFileOpenModeEnums](#)
CameraDefsC.h, [279](#)
- [_spinFileOperationSelectorEnums](#)
CameraDefsC.h, [279](#)
- [_spinFileOperationStatusEnums](#)
CameraDefsC.h, [280](#)
- [_spinFileSelectorEnums](#)
CameraDefsC.h, [280](#)
- [_spinGainAutoBalanceEnums](#)
CameraDefsC.h, [280](#)
- [_spinGainAutoEnums](#)
CameraDefsC.h, [282](#)
- [_spinGainSelectorEnums](#)
CameraDefsC.h, [282](#)
- [_spinGevCCPEnums](#)
CameraDefsC.h, [282](#)
- [_spinGevCurrentPhysicalLinkConfigurationEnums](#)
CameraDefsC.h, [283](#)
- [_spinGevGVCPExtendedStatusCodesSelectorEnums](#)
CameraDefsC.h, [283](#)
- [_spinGevGVSPExtendedIDModeEnums](#)
CameraDefsC.h, [283](#)
- [_spinGevIEEE1588ClockAccuracyEnums](#)
CameraDefsC.h, [284](#)
- [_spinGevIEEE1588ModeEnums](#)
CameraDefsC.h, [284](#)
- [_spinGevIEEE1588StatusEnums](#)
CameraDefsC.h, [284](#)
- [_spinGevIPConfigurationStatusEnums](#)
CameraDefsC.h, [285](#)
- [_spinGevPhysicalLinkConfigurationEnums](#)
CameraDefsC.h, [285](#)
- [_spinGevSupportedOptionSelectorEnums](#)
CameraDefsC.h, [285](#)
- [_spinH264Option](#), [198](#)
 [bitrate](#), [198](#)

- frameRate, [198](#)
- height, [199](#)
- reserved, [199](#)
- width, [199](#)
- _spinImageComponentSelectorEnums
 - CameraDefsC.h, [286](#)
- _spinImageCompressionJPEGFormatOptionEnums
 - CameraDefsC.h, [287](#)
- _spinImageCompressionModeEnums
 - CameraDefsC.h, [287](#)
- _spinImageCompressionRateOptionEnums
 - CameraDefsC.h, [288](#)
- _spinImageFileFormat
 - SpinnakerDefsC.h, [448](#)
- _spinImageStatus
 - SpinnakerDefsC.h, [448](#)
- _spinIncMode
 - SpinnakerGenApiDefsC.h, [498](#)
- _spinInputDirection
 - SpinnakerGenApiDefsC.h, [498](#)
- _spinInterfaceType
 - SpinnakerGenApiDefsC.h, [498](#)
- _spinJPEGOption, [199](#)
 - progressive, [200](#)
 - quality, [200](#)
 - reserved, [200](#)
- _spinJPG2Option, [200](#)
 - quality, [201](#)
 - reserved, [201](#)
- _spinLUTSelectorEnums
 - CameraDefsC.h, [292](#)
- _spinLibraryVersion, [201](#)
 - build, [202](#)
 - major, [202](#)
 - minor, [202](#)
 - type, [202](#)
- _spinLineFormatEnums
 - CameraDefsC.h, [288](#)
- _spinLineInputFilterSelectorEnums
 - CameraDefsC.h, [288](#)
- _spinLineModeEnums
 - CameraDefsC.h, [289](#)
- _spinLineSelectorEnums
 - CameraDefsC.h, [289](#)
- _spinLineSourceEnums
 - CameraDefsC.h, [289](#)
- _spinLinkType
 - SpinnakerGenApiDefsC.h, [499](#)
- _spinLogLevel
 - SpinnakerDefsC.h, [449](#)
- _spinLogicBlockLUTInputActivationEnums
 - CameraDefsC.h, [290](#)
- _spinLogicBlockLUTInputSelectorEnums
 - CameraDefsC.h, [290](#)
- _spinLogicBlockLUTInputSourceEnums
 - CameraDefsC.h, [291](#)
- _spinLogicBlockLUTSelectorEnums
 - CameraDefsC.h, [291](#)
- _spinLogicBlockSelectorEnums
 - CameraDefsC.h, [292](#)
- _spinMJPGOption, [202](#)
 - frameRate, [203](#)
 - height, [203](#)
 - quality, [203](#)
 - reserved, [203](#)
 - width, [204](#)
- _spinNameSpace
 - SpinnakerGenApiDefsC.h, [500](#)
- _spinNodeType
 - SpinnakerGenApiDefsC.h, [500](#)
- _spinPGMOption, [204](#)
 - binaryFile, [204](#)
 - reserved, [204](#)
- _spinPNGOption, [205](#)
 - compressionLevel, [205](#)
 - interlaced, [205](#)
 - reserved, [205](#)
- _spinPPMOption, [206](#)
 - binaryFile, [206](#)
 - reserved, [206](#)
- _spinPixelColorFilterEnums
 - CameraDefsC.h, [292](#)
- _spinPixelFormatEnums
 - CameraDefsC.h, [293](#)
- _spinPixelFormatInfoSelectorEnums
 - CameraDefsC.h, [298](#)
- _spinPixelSizeEnums
 - CameraDefsC.h, [304](#)
- _spinRegionDestinationEnums
 - CameraDefsC.h, [305](#)
- _spinRegionModeEnums
 - CameraDefsC.h, [305](#)
- _spinRegionSelectorEnums
 - CameraDefsC.h, [305](#)
- _spinRepresentation
 - SpinnakerGenApiDefsC.h, [501](#)
- _spinRgbTransformLightSourceEnums
 - CameraDefsC.h, [306](#)
- _spinScan3dCoordinateReferenceSelectorEnums
 - CameraDefsC.h, [306](#)
- _spinScan3dCoordinateSelectorEnums
 - CameraDefsC.h, [307](#)
- _spinScan3dCoordinateSystemEnums
 - CameraDefsC.h, [307](#)
- _spinScan3dCoordinateSystemReferenceEnums
 - CameraDefsC.h, [307](#)
- _spinScan3dCoordinateTransformSelectorEnums
 - CameraDefsC.h, [308](#)
- _spinScan3dDistanceUnitEnums
 - CameraDefsC.h, [308](#)
- _spinScan3dOutputModeEnums
 - CameraDefsC.h, [308](#)
- _spinSensorDigitizationTapsEnums
 - CameraDefsC.h, [309](#)
- _spinSensorShutterModeEnums
 - CameraDefsC.h, [310](#)

- `_spinSensorTapsEnums`
 - `CameraDefsC.h`, [310](#)
- `_spinSequencerConfigurationModeEnums`
 - `CameraDefsC.h`, [310](#)
- `_spinSequencerConfigurationValidEnums`
 - `CameraDefsC.h`, [311](#)
- `_spinSequencerModeEnums`
 - `CameraDefsC.h`, [311](#)
- `_spinSequencerSetValidEnums`
 - `CameraDefsC.h`, [311](#)
- `_spinSequencerTriggerActivationEnums`
 - `CameraDefsC.h`, [312](#)
- `_spinSequencerTriggerSourceEnums`
 - `CameraDefsC.h`, [312](#)
- `_spinSerialPortBaudRateEnums`
 - `CameraDefsC.h`, [312](#)
- `_spinSerialPortParityEnums`
 - `CameraDefsC.h`, [313](#)
- `_spinSerialPortSelectorEnums`
 - `CameraDefsC.h`, [313](#)
- `_spinSerialPortSourceEnums`
 - `CameraDefsC.h`, [314](#)
- `_spinSerialPortStopBitsEnums`
 - `CameraDefsC.h`, [314](#)
- `_spinSign`
 - `SpinnakerGenApiDefsC.h`, [501](#)
- `_spinSlope`
 - `SpinnakerGenApiDefsC.h`, [501](#)
- `_spinSoftwareSignalSelectorEnums`
 - `CameraDefsC.h`, [314](#)
- `_spinSourceSelectorEnums`
 - `CameraDefsC.h`, [315](#)
- `_spinStandardNameSpace`
 - `SpinnakerGenApiDefsC.h`, [502](#)
- `_spinStatisticsChannel`
 - `SpinnakerDefsC.h`, [449](#)
- `_spinTIFFCompressionMethod`
 - `SpinnakerDefsC.h`, [450](#)
- `_spinTIFFOption`, [207](#)
 - `compression`, [207](#)
 - `reserved`, [207](#)
- `_spinTLDeviceAccessStatusEnums`
 - `TransportLayerDefsC.h`, [509](#)
- `_spinTLDeviceCurrentSpeedEnums`
 - `TransportLayerDefsC.h`, [509](#)
- `_spinTLDeviceEndiannessMechanismEnums`
 - `TransportLayerDefsC.h`, [509](#)
- `_spinTLDeviceTypeEnums`
 - `TransportLayerDefsC.h`, [511](#)
- `_spinTLFilterDriverStatusEnums`
 - `TransportLayerDefsC.h`, [511](#)
- `_spinTLGUIXMLLocationEnums`
 - `TransportLayerDefsC.h`, [512](#)
- `_spinTLGenICamXMLLocationEnums`
 - `TransportLayerDefsC.h`, [511](#)
- `_spinTLGevCCPEnums`
 - `TransportLayerDefsC.h`, [512](#)
- `_spinTLInterfaceTypeEnums`
 - `TransportLayerDefsC.h`, [512](#)
- `_spinTLPOEStatusEnums`
 - `TransportLayerDefsC.h`, [513](#)
- `_spinTLPayloadType`
 - `SpinnakerDefsC.h`, [450](#)
- `_spinTLPixelFormatNamespace`
 - `SpinnakerDefsC.h`, [451](#)
- `_spinTLStreamBufferCountModeEnums`
 - `TransportLayerDefsC.h`, [513](#)
- `_spinTLStreamBufferHandlingModeEnums`
 - `TransportLayerDefsC.h`, [513](#)
- `_spinTLStreamModeEnums`
 - `TransportLayerDefsC.h`, [514](#)
- `_spinTLStreamTypeEnums`
 - `TransportLayerDefsC.h`, [514](#)
- `_spinTLTLTypeEnums`
 - `TransportLayerDefsC.h`, [515](#)
- `_spinTestPatternEnums`
 - `CameraDefsC.h`, [315](#)
- `_spinTestPatternGeneratorSelectorEnums`
 - `CameraDefsC.h`, [315](#)
- `_spinTimerSelectorEnums`
 - `CameraDefsC.h`, [316](#)
- `_spinTimerStatusEnums`
 - `CameraDefsC.h`, [316](#)
- `_spinTimerTriggerActivationEnums`
 - `CameraDefsC.h`, [316](#)
- `_spinTimerTriggerSourceEnums`
 - `CameraDefsC.h`, [317](#)
- `_spinTransferComponentSelectorEnums`
 - `CameraDefsC.h`, [318](#)
- `_spinTransferControlModeEnums`
 - `CameraDefsC.h`, [318](#)
- `_spinTransferOperationModeEnums`
 - `CameraDefsC.h`, [319](#)
- `_spinTransferQueueModeEnums`
 - `CameraDefsC.h`, [319](#)
- `_spinTransferSelectorEnums`
 - `CameraDefsC.h`, [319](#)
- `_spinTransferStatusSelectorEnums`
 - `CameraDefsC.h`, [320](#)
- `_spinTransferTriggerActivationEnums`
 - `CameraDefsC.h`, [320](#)
- `_spinTransferTriggerModeEnums`
 - `CameraDefsC.h`, [320](#)
- `_spinTransferTriggerSelectorEnums`
 - `CameraDefsC.h`, [321](#)
- `_spinTransferTriggerSourceEnums`
 - `CameraDefsC.h`, [321](#)
- `_spinTriggerActivationEnums`
 - `CameraDefsC.h`, [322](#)
- `_spinTriggerModeEnums`
 - `CameraDefsC.h`, [323](#)
- `_spinTriggerOverlapEnums`
 - `CameraDefsC.h`, [323](#)
- `_spinTriggerSelectorEnums`
 - `CameraDefsC.h`, [323](#)
- `_spinTriggerSourceEnums`

- CameraDefsC.h, [323](#)
- _spinUserOutputSelectorEnums
 - CameraDefsC.h, [324](#)
- _spinUserSetDefaultEnums
 - CameraDefsC.h, [324](#)
- _spinUserSetSelectorEnums
 - CameraDefsC.h, [325](#)
- _spinVisibility
 - SpinnakerGenApiDefsC.h, [502](#)
- _spinWhiteClipSelectorEnums
 - CameraDefsC.h, [325](#)
- _spinXMLValidation
 - SpinnakerGenApiDefsC.h, [503](#)
- _spinYesNo
 - SpinnakerGenApiDefsC.h, [503](#)
- AasRoiEnable
 - _quickSpin, [82](#)
- AasRoiHeight
 - _quickSpin, [82](#)
- AasRoiOffsetX
 - _quickSpin, [82](#)
- AasRoiOffsetY
 - _quickSpin, [82](#)
- AasRoiWidth
 - _quickSpin, [82](#)
- AcquisitionAbort
 - _quickSpin, [83](#)
- AcquisitionArm
 - _quickSpin, [83](#)
- AcquisitionBurstFrameCount
 - _quickSpin, [83](#)
- AcquisitionFrameCount
 - _quickSpin, [83](#)
- AcquisitionFrameRate
 - _quickSpin, [83](#)
- AcquisitionFrameRateEnable
 - _quickSpin, [83](#)
- AcquisitionLineRate
 - _quickSpin, [83](#)
- AcquisitionMode
 - _quickSpin, [83](#)
- AcquisitionMode_Continuous
 - CameraDefsC.h, [242](#)
- AcquisitionMode_MultiFrame
 - CameraDefsC.h, [242](#)
- AcquisitionMode_SingleFrame
 - CameraDefsC.h, [242](#)
- AcquisitionResultingFrameRate
 - _quickSpin, [84](#)
- AcquisitionStart
 - _quickSpin, [84](#)
- AcquisitionStatus
 - _quickSpin, [84](#)
- AcquisitionStatusSelector
 - _quickSpin, [84](#)
- AcquisitionStatusSelector_AcquisitionActive
 - CameraDefsC.h, [242](#)
- AcquisitionStatusSelector_AcquisitionTransfer
 - CameraDefsC.h, [242](#)
- AcquisitionStatusSelector_AcquisitionTriggerWait
 - CameraDefsC.h, [242](#)
- AcquisitionStatusSelector_ExposureActive
 - CameraDefsC.h, [242](#)
- AcquisitionStatusSelector_FrameActive
 - CameraDefsC.h, [242](#)
- AcquisitionStatusSelector_FrameTriggerWait
 - CameraDefsC.h, [242](#)
- AcquisitionStop
 - _quickSpin, [84](#)
- ActionCommand
 - _quickSpinTLInterface, [174](#)
- ActionDeviceKey
 - _quickSpin, [84](#)
- ActionGroupKey
 - _quickSpin, [84](#)
- ActionGroupMask
 - _quickSpin, [84](#)
- ActionQueueSize
 - _quickSpin, [85](#)
- ActionSelector
 - _quickSpin, [85](#)
- ActionUnconditionalMode
 - _quickSpin, [85](#)
- ActionUnconditionalMode_Off
 - CameraDefsC.h, [243](#)
- ActionUnconditionalMode_On
 - CameraDefsC.h, [243](#)
- AdaptiveCompressionEnable
 - _quickSpin, [85](#)
- AdcBitDepth
 - _quickSpin, [85](#)
- AdcBitDepth_Bit10
 - CameraDefsC.h, [243](#)
- AdcBitDepth_Bit12
 - CameraDefsC.h, [243](#)
- AdcBitDepth_Bit14
 - CameraDefsC.h, [243](#)
- AdcBitDepth_Bit8
 - CameraDefsC.h, [243](#)
- aPAUSEMACCtrlFramesReceived
 - _quickSpin, [85](#)
- aPAUSEMACCtrlFramesTransmitted
 - _quickSpin, [85](#)
- AutoAlgorithmSelector
 - _quickSpin, [85](#)
- AutoAlgorithmSelector_Ae
 - CameraDefsC.h, [243](#)
- AutoAlgorithmSelector_Awb
 - CameraDefsC.h, [243](#)
- AutoExposureControlLoopDamping
 - _quickSpin, [86](#)
- AutoExposureControlPriority
 - _quickSpin, [86](#)
- AutoExposureControlPriority_ExposureTime
 - CameraDefsC.h, [244](#)
- AutoExposureControlPriority_Gain

- CameraDefsC.h, [244](#)
- AutoExposureEVCompensation
 - _quickSpin, [86](#)
- AutoExposureExposureTimeLowerLimit
 - _quickSpin, [86](#)
- AutoExposureExposureTimeUpperLimit
 - _quickSpin, [86](#)
- AutoExposureGainLowerLimit
 - _quickSpin, [86](#)
- AutoExposureGainUpperLimit
 - _quickSpin, [86](#)
- AutoExposureGreyValueLowerLimit
 - _quickSpin, [86](#)
- AutoExposureGreyValueUpperLimit
 - _quickSpin, [87](#)
- AutoExposureLightingMode
 - _quickSpin, [87](#)
- AutoExposureLightingMode_AutoDetect
 - CameraDefsC.h, [244](#)
- AutoExposureLightingMode_Backlight
 - CameraDefsC.h, [244](#)
- AutoExposureLightingMode_Frontlight
 - CameraDefsC.h, [244](#)
- AutoExposureLightingMode_Normal
 - CameraDefsC.h, [244](#)
- AutoExposureMeteringMode
 - _quickSpin, [87](#)
- AutoExposureMeteringMode_Average
 - CameraDefsC.h, [245](#)
- AutoExposureMeteringMode_CenterWeighted
 - CameraDefsC.h, [245](#)
- AutoExposureMeteringMode_HistogramPeak
 - CameraDefsC.h, [245](#)
- AutoExposureMeteringMode_Partial
 - CameraDefsC.h, [245](#)
- AutoExposureMeteringMode_Spot
 - CameraDefsC.h, [245](#)
- AutoExposureTargetGreyValue
 - _quickSpin, [87](#)
- AutoExposureTargetGreyValueAuto
 - _quickSpin, [87](#)
- AutoExposureTargetGreyValueAuto_Continuous
 - CameraDefsC.h, [245](#)
- AutoExposureTargetGreyValueAuto_Off
 - CameraDefsC.h, [245](#)
- Automatic
 - SpinnakerGenApiDefsC.h, [502](#)
- BalanceRatio
 - _quickSpin, [87](#)
- BalanceRatioSelector
 - _quickSpin, [87](#)
- BalanceRatioSelector_Blue
 - CameraDefsC.h, [245](#)
- BalanceRatioSelector_Red
 - CameraDefsC.h, [245](#)
- BalanceWhiteAuto
 - _quickSpin, [87](#)
- BalanceWhiteAuto_Continuous
 - CameraDefsC.h, [246](#)
- BalanceWhiteAuto_Off
 - CameraDefsC.h, [246](#)
- BalanceWhiteAuto_Once
 - CameraDefsC.h, [246](#)
- BalanceWhiteAutoDamping
 - _quickSpin, [88](#)
- BalanceWhiteAutoLowerLimit
 - _quickSpin, [88](#)
- BalanceWhiteAutoProfile
 - _quickSpin, [88](#)
- BalanceWhiteAutoProfile_Indoor
 - CameraDefsC.h, [246](#)
- BalanceWhiteAutoProfile_Outdoor
 - CameraDefsC.h, [246](#)
- BalanceWhiteAutoUpperLimit
 - _quickSpin, [88](#)
- BaseNode
 - SpinnakerGenApiDefsC.h, [500](#)
- Beginner
 - SpinnakerGenApiDefsC.h, [502](#)
- BigEndian
 - SpinnakerGenApiDefsC.h, [498](#)
- binaryFile
 - _spinPGMOption, [204](#)
 - _spinPPMOption, [206](#)
- BinningHorizontal
 - _quickSpin, [88](#)
- BinningHorizontalMode
 - _quickSpin, [88](#)
- BinningHorizontalMode_Average
 - CameraDefsC.h, [246](#)
- BinningHorizontalMode_Sum
 - CameraDefsC.h, [246](#)
- BinningSelector
 - _quickSpin, [88](#)
- BinningSelector_All
 - CameraDefsC.h, [247](#)
- BinningSelector_ISP
 - CameraDefsC.h, [247](#)
- BinningSelector_Sensor
 - CameraDefsC.h, [247](#)
- BinningVertical
 - _quickSpin, [88](#)
- BinningVerticalMode
 - _quickSpin, [89](#)
- BinningVerticalMode_Average
 - CameraDefsC.h, [247](#)
- BinningVerticalMode_Sum
 - CameraDefsC.h, [247](#)
- bitrate
 - _spinH264Option, [198](#)
- BlackLevel
 - _quickSpin, [89](#)
- BlackLevelAuto
 - _quickSpin, [89](#)
- BlackLevelAuto_Continuous
 - CameraDefsC.h, [248](#)

- BlackLevelAuto_Off
 - CameraDefsC.h, [248](#)
- BlackLevelAuto_Once
 - CameraDefsC.h, [248](#)
- BlackLevelAutoBalance
 - _quickSpin, [89](#)
- BlackLevelAutoBalance_Continuous
 - CameraDefsC.h, [247](#)
- BlackLevelAutoBalance_Off
 - CameraDefsC.h, [247](#)
- BlackLevelAutoBalance_Once
 - CameraDefsC.h, [247](#)
- BlackLevelClampingEnable
 - _quickSpin, [89](#)
- BlackLevelRaw
 - _quickSpin, [89](#)
- BlackLevelSelector
 - _quickSpin, [89](#)
- BlackLevelSelector_All
 - CameraDefsC.h, [248](#)
- BlackLevelSelector_Analog
 - CameraDefsC.h, [248](#)
- BlackLevelSelector_Digital
 - CameraDefsC.h, [248](#)
- bool8_t
 - SpinnakerDefsC.h, [441](#)
- Boolean
 - SpinnakerGenApiDefsC.h, [501](#)
- BooleanNode
 - SpinnakerGenApiDefsC.h, [500](#)
- build
 - _spinLibraryVersion, [202](#)
- Camera Access, [33](#)
- Camera Enumerations, [22](#)
- CameraDefsC.h
 - _spinAcquisitionModeEnums, [242](#)
 - _spinAcquisitionStatusSelectorEnums, [242](#)
 - _spinActionUnconditionalModeEnums, [242](#)
 - _spinAdcBitDepthEnums, [243](#)
 - _spinAutoAlgorithmSelectorEnums, [243](#)
 - _spinAutoExposureControlPriorityEnums, [243](#)
 - _spinAutoExposureLightingModeEnums, [244](#)
 - _spinAutoExposureMeteringModeEnums, [244](#)
 - _spinAutoExposureTargetGreyValueAutoEnums, [245](#)
 - _spinBalanceRatioSelectorEnums, [245](#)
 - _spinBalanceWhiteAutoEnums, [245](#)
 - _spinBalanceWhiteAutoProfileEnums, [246](#)
 - _spinBinningHorizontalModeEnums, [246](#)
 - _spinBinningSelectorEnums, [246](#)
 - _spinBinningVerticalModeEnums, [247](#)
 - _spinBlackLevelAutoBalanceEnums, [247](#)
 - _spinBlackLevelAutoEnums, [247](#)
 - _spinBlackLevelSelectorEnums, [248](#)
 - _spinChunkBlackLevelSelectorEnums, [248](#)
 - _spinChunkCounterSelectorEnums, [248](#)
 - _spinChunkEncoderSelectorEnums, [249](#)
 - _spinChunkEncoderStatusEnums, [249](#)
 - _spinChunkExposureTimeSelectorEnums, [249](#)
 - _spinChunkGainSelectorEnums, [250](#)
 - _spinChunkImageComponentEnums, [250](#)
 - _spinChunkPixelFormatEnums, [251](#)
 - _spinChunkRegionIDEnums, [251](#)
 - _spinChunkScan3dCoordinateReferenceSelectorEnums, [251](#)
 - _spinChunkScan3dCoordinateSelectorEnums, [252](#)
 - _spinChunkScan3dCoordinateSystemEnums, [252](#)
 - _spinChunkScan3dCoordinateSystemReferenceEnums, [252](#)
 - _spinChunkScan3dCoordinateTransformSelectorEnums, [253](#)
 - _spinChunkScan3dDistanceUnitEnums, [253](#)
 - _spinChunkScan3dOutputModeEnums, [253](#)
 - _spinChunkSelectorEnums, [254](#)
 - _spinChunkSourceIDEnums, [255](#)
 - _spinChunkTimerSelectorEnums, [255](#)
 - _spinChunkTransferStreamIDEnums, [256](#)
 - _spinCIConfigurationEnums, [256](#)
 - _spinCITimeSlotsCountEnums, [256](#)
 - _spinColorTransformationSelectorEnums, [257](#)
 - _spinColorTransformationValueSelectorEnums, [257](#)
 - _spinCompressionSaturationPriorityEnums, [258](#)
 - _spinCounterEventActivationEnums, [258](#)
 - _spinCounterEventSourceEnums, [258](#)
 - _spinCounterResetActivationEnums, [259](#)
 - _spinCounterResetSourceEnums, [259](#)
 - _spinCounterSelectorEnums, [260](#)
 - _spinCounterStatusEnums, [260](#)
 - _spinCounterTriggerActivationEnums, [261](#)
 - _spinCounterTriggerSourceEnums, [261](#)
 - _spinCxpConnectionTestModeEnums, [262](#)
 - _spinCxpLinkConfigurationEnums, [262](#)
 - _spinCxpLinkConfigurationPreferredEnums, [263](#)
 - _spinCxpLinkConfigurationStatusEnums, [264](#)
 - _spinCxpPoCxpStatusEnums, [265](#)
 - _spinDecimationHorizontalModeEnums, [265](#)
 - _spinDecimationSelectorEnums, [265](#)
 - _spinDecimationVerticalModeEnums, [266](#)
 - _spinDefectCorrectionModeEnums, [266](#)
 - _spinDeinterlacingEnums, [266](#)
 - _spinDeviceCharacterSetEnums, [267](#)
 - _spinDeviceClockSelectorEnums, [267](#)
 - _spinDeviceConnectionStatusEnums, [267](#)
 - _spinDeviceIndicatorModeEnums, [268](#)
 - _spinDeviceLinkHeartbeatModeEnums, [268](#)
 - _spinDeviceLinkThroughputLimitModeEnums, [268](#)
 - _spinDevicePowerSupplySelectorEnums, [268](#)
 - _spinDeviceRegistersEndiannessEnums, [269](#)
 - _spinDeviceScanTypeEnums, [269](#)
 - _spinDeviceSerialPortBaudRateEnums, [269](#)
 - _spinDeviceSerialPortSelectorEnums, [270](#)
 - _spinDeviceStreamChannelEndiannessEnums, [270](#)
 - _spinDeviceStreamChannelTypeEnums, [270](#)
 - _spinDeviceTLTypeEnums, [272](#)

- [_spinDeviceTapGeometryEnums, 271](#)
- [_spinDeviceTemperatureSelectorEnums, 272](#)
- [_spinDeviceTypeEnums, 273](#)
- [_spinEncoderModeEnums, 273](#)
- [_spinEncoderOutputModeEnums, 273](#)
- [_spinEncoderResetActivationEnums, 274](#)
- [_spinEncoderResetSourceEnums, 274](#)
- [_spinEncoderSelectorEnums, 275](#)
- [_spinEncoderSourceAEnums, 276](#)
- [_spinEncoderSourceBEnums, 276](#)
- [_spinEncoderStatusEnums, 276](#)
- [_spinEventNotificationEnums, 277](#)
- [_spinEventSelectorEnums, 277](#)
- [_spinExposureActiveModeEnums, 277](#)
- [_spinExposureAutoEnums, 277](#)
- [_spinExposureModeEnums, 278](#)
- [_spinExposureTimeModeEnums, 278](#)
- [_spinExposureTimeSelectorEnums, 279](#)
- [_spinFileOpenModeEnums, 279](#)
- [_spinFileOperationSelectorEnums, 279](#)
- [_spinFileOperationStatusEnums, 280](#)
- [_spinFileSelectorEnums, 280](#)
- [_spinGainAutoBalanceEnums, 280](#)
- [_spinGainAutoEnums, 282](#)
- [_spinGainSelectorEnums, 282](#)
- [_spinGevCCPEnums, 282](#)
- [_spinGevCurrentPhysicalLinkConfigurationEnums, 283](#)
- [_spinGevGVCPExtendedStatusCodesSelectorEnums, 283](#)
- [_spinGevGVSPExtendedIDModeEnums, 283](#)
- [_spinGevIEEE1588ClockAccuracyEnums, 284](#)
- [_spinGevIEEE1588ModeEnums, 284](#)
- [_spinGevIEEE1588StatusEnums, 284](#)
- [_spinGevIPConfigurationStatusEnums, 285](#)
- [_spinGevPhysicalLinkConfigurationEnums, 285](#)
- [_spinGevSupportedOptionSelectorEnums, 285](#)
- [_spinImageComponentSelectorEnums, 286](#)
- [_spinImageCompressionJPEGFormatOptionEnums, 287](#)
- [_spinImageCompressionModeEnums, 287](#)
- [_spinImageCompressionRateOptionEnums, 288](#)
- [_spinLUTSelectorEnums, 292](#)
- [_spinLineFormatEnums, 288](#)
- [_spinLineInputFilterSelectorEnums, 288](#)
- [_spinLineModeEnums, 289](#)
- [_spinLineSelectorEnums, 289](#)
- [_spinLineSourceEnums, 289](#)
- [_spinLogicBlockLUTInputActivationEnums, 290](#)
- [_spinLogicBlockLUTInputSelectorEnums, 290](#)
- [_spinLogicBlockLUTInputSourceEnums, 291](#)
- [_spinLogicBlockLUTSelectorEnums, 291](#)
- [_spinLogicBlockSelectorEnums, 292](#)
- [_spinPixelColorFilterEnums, 292](#)
- [_spinPixelFormatEnums, 293](#)
- [_spinPixelFormatInfoSelectorEnums, 298](#)
- [_spinPixelSizeEnums, 304](#)
- [_spinRegionDestinationEnums, 305](#)
- [_spinRegionModeEnums, 305](#)
- [_spinRegionSelectorEnums, 305](#)
- [_spinRgbTransformLightSourceEnums, 306](#)
- [_spinScan3dCoordinateReferenceSelectorEnums, 306](#)
- [_spinScan3dCoordinateSelectorEnums, 307](#)
- [_spinScan3dCoordinateSystemEnums, 307](#)
- [_spinScan3dCoordinateSystemReferenceEnums, 307](#)
- [_spinScan3dCoordinateTransformSelectorEnums, 308](#)
- [_spinScan3dDistanceUnitEnums, 308](#)
- [_spinScan3dOutputModeEnums, 308](#)
- [_spinSensorDigitizationTapsEnums, 309](#)
- [_spinSensorShutterModeEnums, 310](#)
- [_spinSensorTapsEnums, 310](#)
- [_spinSequencerConfigurationModeEnums, 310](#)
- [_spinSequencerConfigurationValidEnums, 311](#)
- [_spinSequencerModeEnums, 311](#)
- [_spinSequencerSetValidEnums, 311](#)
- [_spinSequencerTriggerActivationEnums, 312](#)
- [_spinSequencerTriggerSourceEnums, 312](#)
- [_spinSerialPortBaudRateEnums, 312](#)
- [_spinSerialPortParityEnums, 313](#)
- [_spinSerialPortSelectorEnums, 313](#)
- [_spinSerialPortSourceEnums, 314](#)
- [_spinSerialPortStopBitsEnums, 314](#)
- [_spinSoftwareSignalSelectorEnums, 314](#)
- [_spinSourceSelectorEnums, 315](#)
- [_spinTestPatternEnums, 315](#)
- [_spinTestPatternGeneratorSelectorEnums, 315](#)
- [_spinTimerSelectorEnums, 316](#)
- [_spinTimerStatusEnums, 316](#)
- [_spinTimerTriggerActivationEnums, 316](#)
- [_spinTimerTriggerSourceEnums, 317](#)
- [_spinTransferComponentSelectorEnums, 318](#)
- [_spinTransferControlModeEnums, 318](#)
- [_spinTransferOperationModeEnums, 319](#)
- [_spinTransferQueueModeEnums, 319](#)
- [_spinTransferSelectorEnums, 319](#)
- [_spinTransferStatusSelectorEnums, 320](#)
- [_spinTransferTriggerActivationEnums, 320](#)
- [_spinTransferTriggerModeEnums, 320](#)
- [_spinTransferTriggerSelectorEnums, 321](#)
- [_spinTransferTriggerSourceEnums, 321](#)
- [_spinTriggerActivationEnums, 322](#)
- [_spinTriggerModeEnums, 323](#)
- [_spinTriggerOverlapEnums, 323](#)
- [_spinTriggerSelectorEnums, 323](#)
- [_spinTriggerSourceEnums, 323](#)
- [_spinUserOutputSelectorEnums, 324](#)
- [_spinUserSetDefaultEnums, 324](#)
- [_spinUserSetSelectorEnums, 325](#)
- [_spinWhiteClipSelectorEnums, 325](#)
- [AcquisitionMode_Continuous, 242](#)
- [AcquisitionMode_MultiFrame, 242](#)
- [AcquisitionMode_SingleFrame, 242](#)
- [AcquisitionStatusSelector_AcquisitionActive, 242](#)

- AcquisitionStatusSelector_AcquisitionTransfer, [242](#)
- AcquisitionStatusSelector_AcquisitionTriggerWait, [242](#)
- AcquisitionStatusSelector_ExposureActive, [242](#)
- AcquisitionStatusSelector_FrameActive, [242](#)
- AcquisitionStatusSelector_FrameTriggerWait, [242](#)
- ActionUnconditionalMode_Off, [243](#)
- ActionUnconditionalMode_On, [243](#)
- AdcBitDepth_Bit10, [243](#)
- AdcBitDepth_Bit12, [243](#)
- AdcBitDepth_Bit14, [243](#)
- AdcBitDepth_Bit8, [243](#)
- AutoAlgorithmSelector_Ae, [243](#)
- AutoAlgorithmSelector_Awb, [243](#)
- AutoExposureControlPriority_ExposureTime, [244](#)
- AutoExposureControlPriority_Gain, [244](#)
- AutoExposureLightingMode_AutoDetect, [244](#)
- AutoExposureLightingMode_Backlight, [244](#)
- AutoExposureLightingMode_Frontlight, [244](#)
- AutoExposureLightingMode_Normal, [244](#)
- AutoExposureMeteringMode_Average, [245](#)
- AutoExposureMeteringMode_CenterWeighted, [245](#)
- AutoExposureMeteringMode_HistogramPeak, [245](#)
- AutoExposureMeteringMode_Partial, [245](#)
- AutoExposureMeteringMode_Spot, [245](#)
- AutoExposureTargetGreyValueAuto_Continuous, [245](#)
- AutoExposureTargetGreyValueAuto_Off, [245](#)
- BalanceRatioSelector_Blue, [245](#)
- BalanceRatioSelector_Red, [245](#)
- BalanceWhiteAuto_Continuous, [246](#)
- BalanceWhiteAuto_Off, [246](#)
- BalanceWhiteAuto_Once, [246](#)
- BalanceWhiteAutoProfile_Indoor, [246](#)
- BalanceWhiteAutoProfile_Outdoor, [246](#)
- BinningHorizontalMode_Average, [246](#)
- BinningHorizontalMode_Sum, [246](#)
- BinningSelector_All, [247](#)
- BinningSelector_ISP, [247](#)
- BinningSelector_Sensor, [247](#)
- BinningVerticalMode_Average, [247](#)
- BinningVerticalMode_Sum, [247](#)
- BlackLevelAuto_Continuous, [248](#)
- BlackLevelAuto_Off, [248](#)
- BlackLevelAuto_Once, [248](#)
- BlackLevelAutoBalance_Continuous, [247](#)
- BlackLevelAutoBalance_Off, [247](#)
- BlackLevelAutoBalance_Once, [247](#)
- BlackLevelSelector_All, [248](#)
- BlackLevelSelector_Analog, [248](#)
- BlackLevelSelector_Digital, [248](#)
- ChunkBlackLevelSelector_All, [248](#)
- ChunkCounterSelector_Counter0, [249](#)
- ChunkCounterSelector_Counter1, [249](#)
- ChunkCounterSelector_Counter2, [249](#)
- ChunkEncoderSelector_Encoder0, [249](#)
- ChunkEncoderSelector_Encoder1, [249](#)
- ChunkEncoderSelector_Encoder2, [249](#)
- ChunkEncoderStatus_EncoderDown, [249](#)
- ChunkEncoderStatus_EncoderIdle, [249](#)
- ChunkEncoderStatus_EncoderStatic, [249](#)
- ChunkEncoderStatus_EncoderUp, [249](#)
- ChunkExposureTimeSelector_Blue, [250](#)
- ChunkExposureTimeSelector_Common, [250](#)
- ChunkExposureTimeSelector_Cyan, [250](#)
- ChunkExposureTimeSelector_Green, [250](#)
- ChunkExposureTimeSelector_Infrared, [250](#)
- ChunkExposureTimeSelector_Magenta, [250](#)
- ChunkExposureTimeSelector_Red, [250](#)
- ChunkExposureTimeSelector_Stage1, [250](#)
- ChunkExposureTimeSelector_Stage2, [250](#)
- ChunkExposureTimeSelector_Ultraviolet, [250](#)
- ChunkExposureTimeSelector_Yellow, [250](#)
- ChunkGainSelector_All, [250](#)
- ChunkGainSelector_Blue, [250](#)
- ChunkGainSelector_Green, [250](#)
- ChunkGainSelector_Red, [250](#)
- ChunkImageComponent_Color, [250](#)
- ChunkImageComponent_Confidence, [251](#)
- ChunkImageComponent_Disparity, [250](#)
- ChunkImageComponent_Infrared, [250](#)
- ChunkImageComponent_Intensity, [250](#)
- ChunkImageComponent_Range, [250](#)
- ChunkImageComponent_Scatter, [251](#)
- ChunkImageComponent_Ultraviolet, [250](#)
- ChunkPixelFormat_BayerBG8, [251](#)
- ChunkPixelFormat_BayerGB8, [251](#)
- ChunkPixelFormat_BayerGR8, [251](#)
- ChunkPixelFormat_BayerRG8, [251](#)
- ChunkPixelFormat_Mono12Packed, [251](#)
- ChunkPixelFormat_Mono16, [251](#)
- ChunkPixelFormat_Mono8, [251](#)
- ChunkPixelFormat_RGB8Packed, [251](#)
- ChunkPixelFormat_YCbCr601_422_8_CbYCrY, [251](#)
- ChunkPixelFormat_YUV422Packed, [251](#)
- ChunkRegionID_Region0, [251](#)
- ChunkRegionID_Region1, [251](#)
- ChunkRegionID_Region2, [251](#)
- ChunkScan3dCoordinateReferenceSelector_RotationX, [252](#)
- ChunkScan3dCoordinateReferenceSelector_RotationY, [252](#)
- ChunkScan3dCoordinateReferenceSelector_RotationZ, [252](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationX, [252](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationY, [252](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationZ, [252](#)
- ChunkScan3dCoordinateSelector_CoordinateA, [252](#)
- ChunkScan3dCoordinateSelector_CoordinateB, [252](#)

- ChunkScan3dCoordinateSelector_CoordinateC, [252](#)
- ChunkScan3dCoordinateSystem_Cartesian, [252](#)
- ChunkScan3dCoordinateSystem_Cylindrical, [252](#)
- ChunkScan3dCoordinateSystem_Spherical, [252](#)
- ChunkScan3dCoordinateSystemReference_Anchor, [253](#)
- ChunkScan3dCoordinateSystemReference_Transformed, [253](#)
- ChunkScan3dCoordinateTransformSelector_RotationX, [253](#)
- ChunkScan3dCoordinateTransformSelector_RotationY, [253](#)
- ChunkScan3dCoordinateTransformSelector_RotationZ, [253](#)
- ChunkScan3dCoordinateTransformSelector_TranslationX, [253](#)
- ChunkScan3dCoordinateTransformSelector_TranslationY, [253](#)
- ChunkScan3dCoordinateTransformSelector_TranslationZ, [253](#)
- ChunkScan3dDistanceUnit_Inch, [253](#)
- ChunkScan3dDistanceUnit_Millimeter, [253](#)
- ChunkScan3dOutputMode_CalibratedABC_Grid, [254](#)
- ChunkScan3dOutputMode_CalibratedABC_PointCloud, [254](#)
- ChunkScan3dOutputMode_CalibratedAC, [254](#)
- ChunkScan3dOutputMode_CalibratedAC_Linescan, [254](#)
- ChunkScan3dOutputMode_CalibratedC, [254](#)
- ChunkScan3dOutputMode_CalibratedC_Linescan, [254](#)
- ChunkScan3dOutputMode_DisparityC, [254](#)
- ChunkScan3dOutputMode_DisparityC_Linescan, [254](#)
- ChunkScan3dOutputMode_RectifiedC, [254](#)
- ChunkScan3dOutputMode_RectifiedC_Linescan, [254](#)
- ChunkScan3dOutputMode_UncalibratedC, [254](#)
- ChunkSelector_BlackLevel, [255](#)
- ChunkSelector_CRC, [255](#)
- ChunkSelector_ExposureEndLineStatusAll, [255](#)
- ChunkSelector_ExposureTime, [255](#)
- ChunkSelector_FrameID, [255](#)
- ChunkSelector_Gain, [255](#)
- ChunkSelector_Height, [255](#)
- ChunkSelector_Image, [255](#)
- ChunkSelector_OffsetX, [255](#)
- ChunkSelector_OffsetY, [255](#)
- ChunkSelector_PixelFormat, [255](#)
- ChunkSelector_SequencerSetActive, [255](#)
- ChunkSelector_SerialData, [255](#)
- ChunkSelector_Timestamp, [255](#)
- ChunkSelector_Width, [255](#)
- ChunkSourceID_Source0, [255](#)
- ChunkSourceID_Source1, [255](#)
- ChunkSourceID_Source2, [255](#)
- ChunkTimerSelector_Timer0, [256](#)
- ChunkTimerSelector_Timer1, [256](#)
- ChunkTimerSelector_Timer2, [256](#)
- ChunkTransferStreamID_Stream0, [256](#)
- ChunkTransferStreamID_Stream1, [256](#)
- ChunkTransferStreamID_Stream2, [256](#)
- ChunkTransferStreamID_Stream3, [256](#)
- CIConfiguration_Base, [256](#)
- CIConfiguration_DualBase, [256](#)
- CIConfiguration_EightyBit, [256](#)
- CIConfiguration_Full, [256](#)
- CIConfiguration_Medium, [256](#)
- CITimeSlotsCount_One, [257](#)
- CITimeSlotsCount_Three, [257](#)
- CITimeSlotsCount_Two, [257](#)
- ColorTransformationSelector_RGBtoRGB, [257](#)
- ColorTransformationSelector_RGBtoYUV, [257](#)
- ColorTransformationValueSelector_Gain00, [257](#)
- ColorTransformationValueSelector_Gain01, [257](#)
- ColorTransformationValueSelector_Gain02, [257](#)
- ColorTransformationValueSelector_Gain10, [257](#)
- ColorTransformationValueSelector_Gain11, [257](#)
- ColorTransformationValueSelector_Gain12, [257](#)
- ColorTransformationValueSelector_Gain20, [257](#)
- ColorTransformationValueSelector_Gain21, [257](#)
- ColorTransformationValueSelector_Gain22, [257](#)
- ColorTransformationValueSelector_Offset0, [257](#)
- ColorTransformationValueSelector_Offset1, [257](#)
- ColorTransformationValueSelector_Offset2, [257](#)
- CompressionSaturationPriority_DropFrame, [258](#)
- CompressionSaturationPriority_ReduceFrameRate, [258](#)
- CounterEventActivation_AnyEdge, [258](#)
- CounterEventActivation_FallingEdge, [258](#)
- CounterEventActivation_LevelHigh, [258](#)
- CounterEventActivation_LevelLow, [258](#)
- CounterEventActivation_RisingEdge, [258](#)
- CounterEventSource_Counter0End, [259](#)
- CounterEventSource_Counter0Start, [259](#)
- CounterEventSource_Counter1End, [259](#)
- CounterEventSource_Counter1Start, [259](#)
- CounterEventSource_ExposureEnd, [259](#)
- CounterEventSource_ExposureStart, [259](#)
- CounterEventSource_FrameTriggerWait, [259](#)
- CounterEventSource_Line0, [258](#)
- CounterEventSource_Line1, [259](#)
- CounterEventSource_Line2, [259](#)
- CounterEventSource_Line3, [259](#)
- CounterEventSource_LogicBlock0, [259](#)
- CounterEventSource_LogicBlock1, [259](#)
- CounterEventSource_MHzTick, [258](#)
- CounterEventSource_Off, [258](#)
- CounterEventSource_UserOutput0, [259](#)
- CounterEventSource_UserOutput1, [259](#)
- CounterEventSource_UserOutput2, [259](#)
- CounterEventSource_UserOutput3, [259](#)
- CounterResetActivation_AnyEdge, [259](#)
- CounterResetActivation_FallingEdge, [259](#)

- CounterResetActivation_LevelHigh, 259
- CounterResetActivation_LevelLow, 259
- CounterResetActivation_RisingEdge, 259
- CounterResetSource_Counter0End, 260
- CounterResetSource_Counter0Start, 260
- CounterResetSource_Counter1End, 260
- CounterResetSource_Counter1Start, 260
- CounterResetSource_ExposureEnd, 260
- CounterResetSource_ExposureStart, 260
- CounterResetSource_FrameTriggerWait, 260
- CounterResetSource_Line0, 260
- CounterResetSource_Line1, 260
- CounterResetSource_Line2, 260
- CounterResetSource_Line3, 260
- CounterResetSource_LogicBlock0, 260
- CounterResetSource_LogicBlock1, 260
- CounterResetSource_Off, 259
- CounterResetSource_UserOutput0, 260
- CounterResetSource_UserOutput1, 260
- CounterResetSource_UserOutput2, 260
- CounterResetSource_UserOutput3, 260
- CounterSelector_Counter0, 260
- CounterSelector_Counter1, 260
- CounterStatus_CounterActive, 260
- CounterStatus_CounterCompleted, 260
- CounterStatus_CounterIdle, 260
- CounterStatus_CounterOverflow, 260
- CounterStatus_CounterTriggerWait, 260
- CounterTriggerActivation_AnyEdge, 261
- CounterTriggerActivation_FallingEdge, 261
- CounterTriggerActivation_LevelHigh, 261
- CounterTriggerActivation_LevelLow, 261
- CounterTriggerActivation_RisingEdge, 261
- CounterTriggerSource_Counter0End, 261
- CounterTriggerSource_Counter0Start, 261
- CounterTriggerSource_Counter1End, 261
- CounterTriggerSource_Counter1Start, 261
- CounterTriggerSource_ExposureEnd, 261
- CounterTriggerSource_ExposureStart, 261
- CounterTriggerSource_FrameTriggerWait, 261
- CounterTriggerSource_Line0, 261
- CounterTriggerSource_Line1, 261
- CounterTriggerSource_Line2, 261
- CounterTriggerSource_Line3, 261
- CounterTriggerSource_LogicBlock0, 261
- CounterTriggerSource_LogicBlock1, 261
- CounterTriggerSource_Off, 261
- CounterTriggerSource_UserOutput0, 261
- CounterTriggerSource_UserOutput1, 261
- CounterTriggerSource_UserOutput2, 261
- CounterTriggerSource_UserOutput3, 261
- CxpConnectionTestMode_Mode1, 262
- CxpConnectionTestMode_Off, 262
- CxpLinkConfiguration_Auto, 262
- CxpLinkConfiguration_CXP1_X1, 262
- CxpLinkConfiguration_CXP1_X2, 262
- CxpLinkConfiguration_CXP1_X3, 262
- CxpLinkConfiguration_CXP1_X4, 262
- CxpLinkConfiguration_CXP1_X5, 263
- CxpLinkConfiguration_CXP1_X6, 263
- CxpLinkConfiguration_CXP2_X1, 262
- CxpLinkConfiguration_CXP2_X2, 262
- CxpLinkConfiguration_CXP2_X3, 262
- CxpLinkConfiguration_CXP2_X4, 262
- CxpLinkConfiguration_CXP2_X5, 263
- CxpLinkConfiguration_CXP2_X6, 263
- CxpLinkConfiguration_CXP3_X1, 262
- CxpLinkConfiguration_CXP3_X2, 262
- CxpLinkConfiguration_CXP3_X3, 262
- CxpLinkConfiguration_CXP3_X4, 262
- CxpLinkConfiguration_CXP3_X5, 263
- CxpLinkConfiguration_CXP3_X6, 263
- CxpLinkConfiguration_CXP5_X1, 262
- CxpLinkConfiguration_CXP5_X2, 262
- CxpLinkConfiguration_CXP5_X3, 262
- CxpLinkConfiguration_CXP5_X4, 262
- CxpLinkConfiguration_CXP5_X5, 263
- CxpLinkConfiguration_CXP5_X6, 263
- CxpLinkConfiguration_CXP6_X1, 262
- CxpLinkConfiguration_CXP6_X2, 262
- CxpLinkConfiguration_CXP6_X3, 262
- CxpLinkConfiguration_CXP6_X4, 262
- CxpLinkConfiguration_CXP6_X5, 263
- CxpLinkConfiguration_CXP6_X6, 263
- CxpLinkConfigurationPreferred_CXP1_X1, 263
- CxpLinkConfigurationPreferred_CXP1_X2, 263
- CxpLinkConfigurationPreferred_CXP1_X3, 263
- CxpLinkConfigurationPreferred_CXP1_X4, 263
- CxpLinkConfigurationPreferred_CXP1_X5, 263
- CxpLinkConfigurationPreferred_CXP1_X6, 264
- CxpLinkConfigurationPreferred_CXP2_X1, 263
- CxpLinkConfigurationPreferred_CXP2_X2, 263
- CxpLinkConfigurationPreferred_CXP2_X3, 263
- CxpLinkConfigurationPreferred_CXP2_X4, 263
- CxpLinkConfigurationPreferred_CXP2_X5, 263
- CxpLinkConfigurationPreferred_CXP2_X6, 264
- CxpLinkConfigurationPreferred_CXP3_X1, 263
- CxpLinkConfigurationPreferred_CXP3_X2, 263
- CxpLinkConfigurationPreferred_CXP3_X3, 263
- CxpLinkConfigurationPreferred_CXP3_X4, 263
- CxpLinkConfigurationPreferred_CXP3_X5, 263
- CxpLinkConfigurationPreferred_CXP3_X6, 264
- CxpLinkConfigurationPreferred_CXP5_X1, 263
- CxpLinkConfigurationPreferred_CXP5_X2, 263
- CxpLinkConfigurationPreferred_CXP5_X3, 263
- CxpLinkConfigurationPreferred_CXP5_X4, 263
- CxpLinkConfigurationPreferred_CXP5_X5, 263
- CxpLinkConfigurationPreferred_CXP5_X6, 264
- CxpLinkConfigurationPreferred_CXP6_X1, 263
- CxpLinkConfigurationPreferred_CXP6_X2, 263
- CxpLinkConfigurationPreferred_CXP6_X3, 263
- CxpLinkConfigurationPreferred_CXP6_X4, 263
- CxpLinkConfigurationPreferred_CXP6_X5, 264
- CxpLinkConfigurationPreferred_CXP6_X6, 264
- CxpLinkConfigurationStatus_CXP1_X1, 264
- CxpLinkConfigurationStatus_CXP1_X2, 264

- CxpLinkConfigurationStatus_CXP1_X3, [264](#)
- CxpLinkConfigurationStatus_CXP1_X4, [264](#)
- CxpLinkConfigurationStatus_CXP1_X5, [264](#)
- CxpLinkConfigurationStatus_CXP1_X6, [265](#)
- CxpLinkConfigurationStatus_CXP2_X1, [264](#)
- CxpLinkConfigurationStatus_CXP2_X2, [264](#)
- CxpLinkConfigurationStatus_CXP2_X3, [264](#)
- CxpLinkConfigurationStatus_CXP2_X4, [264](#)
- CxpLinkConfigurationStatus_CXP2_X5, [264](#)
- CxpLinkConfigurationStatus_CXP2_X6, [265](#)
- CxpLinkConfigurationStatus_CXP3_X1, [264](#)
- CxpLinkConfigurationStatus_CXP3_X2, [264](#)
- CxpLinkConfigurationStatus_CXP3_X3, [264](#)
- CxpLinkConfigurationStatus_CXP3_X4, [264](#)
- CxpLinkConfigurationStatus_CXP3_X5, [264](#)
- CxpLinkConfigurationStatus_CXP3_X6, [265](#)
- CxpLinkConfigurationStatus_CXP5_X1, [264](#)
- CxpLinkConfigurationStatus_CXP5_X2, [264](#)
- CxpLinkConfigurationStatus_CXP5_X3, [264](#)
- CxpLinkConfigurationStatus_CXP5_X4, [264](#)
- CxpLinkConfigurationStatus_CXP5_X5, [264](#)
- CxpLinkConfigurationStatus_CXP5_X6, [265](#)
- CxpLinkConfigurationStatus_CXP6_X1, [264](#)
- CxpLinkConfigurationStatus_CXP6_X2, [264](#)
- CxpLinkConfigurationStatus_CXP6_X3, [264](#)
- CxpLinkConfigurationStatus_CXP6_X4, [264](#)
- CxpLinkConfigurationStatus_CXP6_X5, [265](#)
- CxpLinkConfigurationStatus_CXP6_X6, [265](#)
- CxpLinkConfigurationStatus_None, [264](#)
- CxpLinkConfigurationStatus_Pending, [264](#)
- CxpPoCxpStatus_Auto, [265](#)
- CxpPoCxpStatus_Off, [265](#)
- CxpPoCxpStatus_Tripped, [265](#)
- DecimationHorizontalMode_Discard, [265](#)
- DecimationSelector_All, [266](#)
- DecimationSelector_Sensor, [266](#)
- DecimationVerticalMode_Discard, [266](#)
- DefectCorrectionMode_Average, [266](#)
- DefectCorrectionMode_Highlight, [266](#)
- DefectCorrectionMode_Zero, [266](#)
- Deinterlacing_LineDuplication, [267](#)
- Deinterlacing_Off, [267](#)
- Deinterlacing_Weave, [267](#)
- DeviceCharacterSet_ASCII, [267](#)
- DeviceCharacterSet_UTF8, [267](#)
- DeviceClockSelector_CameraLink, [267](#)
- DeviceClockSelector_Sensor, [267](#)
- DeviceClockSelector_SensorDigitization, [267](#)
- DeviceConnectionStatus_Active, [267](#)
- DeviceConnectionStatus_Inactive, [267](#)
- DeviceIndicatorMode_Active, [268](#)
- DeviceIndicatorMode_ErrorStatus, [268](#)
- DeviceIndicatorMode_Inactive, [268](#)
- DeviceLinkHeartbeatMode_Off, [268](#)
- DeviceLinkHeartbeatMode_On, [268](#)
- DeviceLinkThroughputLimitMode_Off, [268](#)
- DeviceLinkThroughputLimitMode_On, [268](#)
- DevicePowerSupplySelector_External, [269](#)
- DeviceRegistersEndianness_Big, [269](#)
- DeviceRegistersEndianness_Little, [269](#)
- DeviceScanType_Areascan, [269](#)
- DeviceSerialPortBaudRate_Baud115200, [270](#)
- DeviceSerialPortBaudRate_Baud19200, [270](#)
- DeviceSerialPortBaudRate_Baud230400, [270](#)
- DeviceSerialPortBaudRate_Baud38400, [270](#)
- DeviceSerialPortBaudRate_Baud460800, [270](#)
- DeviceSerialPortBaudRate_Baud57600, [270](#)
- DeviceSerialPortBaudRate_Baud921600, [270](#)
- DeviceSerialPortBaudRate_Baud9600, [270](#)
- DeviceSerialPortSelector_CameraLink, [270](#)
- DeviceStreamChannelEndianness_Big, [270](#)
- DeviceStreamChannelEndianness_Little, [270](#)
- DeviceStreamChannelType_Receiver, [271](#)
- DeviceStreamChannelType_Transmitter, [271](#)
- DeviceTapGeometry_Geometry_10X, [272](#)
- DeviceTapGeometry_Geometry_10X_1Y, [272](#)
- DeviceTapGeometry_Geometry_1X, [271](#)
- DeviceTapGeometry_Geometry_1X10, [272](#)
- DeviceTapGeometry_Geometry_1X10_1Y, [272](#)
- DeviceTapGeometry_Geometry_1X2, [271](#)
- DeviceTapGeometry_Geometry_1X2_1Y, [271](#)
- DeviceTapGeometry_Geometry_1X2_1Y2, [271](#)
- DeviceTapGeometry_Geometry_1X2_2YE, [271](#)
- DeviceTapGeometry_Geometry_1X3, [271](#)
- DeviceTapGeometry_Geometry_1X3_1Y, [271](#)
- DeviceTapGeometry_Geometry_1X4, [271](#)
- DeviceTapGeometry_Geometry_1X4_1Y, [271](#)
- DeviceTapGeometry_Geometry_1X8, [272](#)
- DeviceTapGeometry_Geometry_1X8_1Y, [272](#)
- DeviceTapGeometry_Geometry_1X_1Y, [271](#)
- DeviceTapGeometry_Geometry_1X_1Y2, [271](#)
- DeviceTapGeometry_Geometry_1X_2YE, [271](#)
- DeviceTapGeometry_Geometry_2X, [271](#)
- DeviceTapGeometry_Geometry_2X2, [271](#)
- DeviceTapGeometry_Geometry_2X2_1Y, [271](#)
- DeviceTapGeometry_Geometry_2X2E, [271](#)
- DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y, [271](#)
- DeviceTapGeometry_Geometry_2X2E_2YE, [272](#)
- DeviceTapGeometry_Geometry_2X2M, [272](#)
- DeviceTapGeometry_Geometry_2X_1Y, [271](#)
- DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y, [271](#)
- DeviceTapGeometry_Geometry_2X_2YE, [271](#)
- DeviceTapGeometry_Geometry_2XE, [271](#)
- DeviceTapGeometry_Geometry_2XE_1Y2, [271](#)
- DeviceTapGeometry_Geometry_2XE_2YE, [271](#)
- DeviceTapGeometry_Geometry_2XM, [271](#)
- DeviceTapGeometry_Geometry_2XM_1Y, [271](#)
- DeviceTapGeometry_Geometry_2XM_1Y2, [271](#)
- DeviceTapGeometry_Geometry_2XM_2YE, [271](#)
- DeviceTapGeometry_Geometry_3X, [271](#)
- DeviceTapGeometry_Geometry_3X_1Y, [271](#)
- DeviceTapGeometry_Geometry_4X, [271](#)
- DeviceTapGeometry_Geometry_4X2, [272](#)
- DeviceTapGeometry_Geometry_4X2_1Y, [272](#)

- DeviceTapGeometry_Geometry_4X2E, [272](#)
- DeviceTapGeometry_Geometry_4X2E_1Y, [272](#)
- DeviceTapGeometry_Geometry_4X_1Y, [271](#)
- DeviceTapGeometry_Geometry_8X, [272](#)
- DeviceTapGeometry_Geometry_8X_1Y, [272](#)
- DeviceTemperatureSelector_Sensor, [272](#)
- DeviceTLType_CameraLink, [272](#)
- DeviceTLType_CameraLinkHS, [272](#)
- DeviceTLType_CoaXPRESS, [272](#)
- DeviceTLType_Custom, [272](#)
- DeviceTLType_GigEVision, [272](#)
- DeviceTLType_USB3Vision, [272](#)
- DeviceType_Peripheral, [273](#)
- DeviceType_Receiver, [273](#)
- DeviceType_Transceiver, [273](#)
- DeviceType_Transmitter, [273](#)
- EncoderMode_FourPhase, [273](#)
- EncoderMode_HighResolution, [273](#)
- EncoderOutputMode_DirectionDown, [274](#)
- EncoderOutputMode_DirectionUp, [274](#)
- EncoderOutputMode_Motion, [274](#)
- EncoderOutputMode_Off, [273](#)
- EncoderOutputMode_PositionDown, [274](#)
- EncoderOutputMode_PositionUp, [273](#)
- EncoderResetActivation_AnyEdge, [274](#)
- EncoderResetActivation_FallingEdge, [274](#)
- EncoderResetActivation_LevelHigh, [274](#)
- EncoderResetActivation_LevelLow, [274](#)
- EncoderResetActivation_RisingEdge, [274](#)
- EncoderResetSource_AcquisitionEnd, [274](#)
- EncoderResetSource_AcquisitionStart, [274](#)
- EncoderResetSource_AcquisitionTrigger, [274](#)
- EncoderResetSource_Action0, [275](#)
- EncoderResetSource_Action1, [275](#)
- EncoderResetSource_Action2, [275](#)
- EncoderResetSource_Counter0End, [275](#)
- EncoderResetSource_Counter0Start, [275](#)
- EncoderResetSource_Counter1End, [275](#)
- EncoderResetSource_Counter1Start, [275](#)
- EncoderResetSource_Counter2End, [275](#)
- EncoderResetSource_Counter2Start, [275](#)
- EncoderResetSource_ExposureEnd, [275](#)
- EncoderResetSource_ExposureStart, [274](#)
- EncoderResetSource_FrameEnd, [274](#)
- EncoderResetSource_FrameStart, [274](#)
- EncoderResetSource_FrameTrigger, [274](#)
- EncoderResetSource_Line0, [275](#)
- EncoderResetSource_Line1, [275](#)
- EncoderResetSource_Line2, [275](#)
- EncoderResetSource_LinkTrigger0, [275](#)
- EncoderResetSource_LinkTrigger1, [275](#)
- EncoderResetSource_LinkTrigger2, [275](#)
- EncoderResetSource_Off, [274](#)
- EncoderResetSource_SoftwareSignal0, [275](#)
- EncoderResetSource_SoftwareSignal1, [275](#)
- EncoderResetSource_SoftwareSignal2, [275](#)
- EncoderResetSource_Timer0End, [275](#)
- EncoderResetSource_Timer0Start, [275](#)
- EncoderResetSource_Timer1End, [275](#)
- EncoderResetSource_Timer1Start, [275](#)
- EncoderResetSource_Timer2End, [275](#)
- EncoderResetSource_Timer2Start, [275](#)
- EncoderResetSource_UserOutput0, [275](#)
- EncoderResetSource_UserOutput1, [275](#)
- EncoderResetSource_UserOutput2, [275](#)
- EncoderSelector_Encoder0, [275](#)
- EncoderSelector_Encoder1, [275](#)
- EncoderSelector_Encoder2, [275](#)
- EncoderSourceA_Line0, [276](#)
- EncoderSourceA_Line1, [276](#)
- EncoderSourceA_Line2, [276](#)
- EncoderSourceA_Off, [276](#)
- EncoderSourceB_Line0, [276](#)
- EncoderSourceB_Line1, [276](#)
- EncoderSourceB_Line2, [276](#)
- EncoderSourceB_Off, [276](#)
- EncoderStatus_EncoderDown, [276](#)
- EncoderStatus_EncoderIdle, [276](#)
- EncoderStatus_EncoderStatic, [276](#)
- EncoderStatus_EncoderUp, [276](#)
- EventNotification_Off, [277](#)
- EventNotification_On, [277](#)
- EventSelector_Error, [277](#)
- EventSelector_ExposureEnd, [277](#)
- EventSelector_SerialPortReceive, [277](#)
- ExposureActiveMode_AllPixels, [277](#)
- ExposureActiveMode_AnyPixels, [277](#)
- ExposureActiveMode_Line1, [277](#)
- ExposureAuto_Continuous, [278](#)
- ExposureAuto_Off, [278](#)
- ExposureAuto_Once, [278](#)
- ExposureMode_Timed, [278](#)
- ExposureMode_TriggerWidth, [278](#)
- ExposureTimeMode_Common, [278](#)
- ExposureTimeMode_Individual, [278](#)
- ExposureTimeSelector_Blue, [279](#)
- ExposureTimeSelector_Common, [279](#)
- ExposureTimeSelector_Cyan, [279](#)
- ExposureTimeSelector_Green, [279](#)
- ExposureTimeSelector_Infrared, [279](#)
- ExposureTimeSelector_Magenta, [279](#)
- ExposureTimeSelector_Red, [279](#)
- ExposureTimeSelector_Stage1, [279](#)
- ExposureTimeSelector_Stage2, [279](#)
- ExposureTimeSelector_Ultraviolet, [279](#)
- ExposureTimeSelector_Yellow, [279](#)
- FileOpenMode_Read, [279](#)
- FileOpenMode_ReadWrite, [279](#)
- FileOpenMode_Write, [279](#)
- FileOperationSelector_Close, [280](#)
- FileOperationSelector_Delete, [280](#)
- FileOperationSelector_Open, [280](#)
- FileOperationSelector_Read, [280](#)
- FileOperationSelector_Write, [280](#)
- FileOperationStatus_Failure, [280](#)
- FileOperationStatus_Overflow, [280](#)

- FileOperationStatus_Success, [280](#)
- FileSelector_SerialPort0, [280](#)
- FileSelector_UserFile1, [280](#)
- FileSelector_UserSet0, [280](#)
- FileSelector_UserSet1, [280](#)
- FileSelector_UserSetDefault, [280](#)
- GainAuto_Continuous, [282](#)
- GainAuto_Off, [282](#)
- GainAuto_Once, [282](#)
- GainAutoBalance_Continuous, [282](#)
- GainAutoBalance_Off, [282](#)
- GainAutoBalance_Once, [282](#)
- GainSelector_All, [282](#)
- GevCCP_ControlAccess, [283](#)
- GevCCP_ExclusiveAccess, [283](#)
- GevCCP_OpenAccess, [283](#)
- GevCurrentPhysicalLinkConfiguration_DynamicLAG, [283](#)
- GevCurrentPhysicalLinkConfiguration_MultiLink, [283](#)
- GevCurrentPhysicalLinkConfiguration_SingleLink, [283](#)
- GevCurrentPhysicalLinkConfiguration_StaticLAG, [283](#)
- GevGVCPExtendedStatusCodesSelector_Version1_1, [283](#)
- GevGVCPExtendedStatusCodesSelector_Version2_0, [283](#)
- GevGVSPExtendedIDMode_Off, [284](#)
- GevGVSPExtendedIDMode_On, [284](#)
- GevIEEE1588ClockAccuracy_Unknown, [284](#)
- GevIEEE1588Mode_Auto, [284](#)
- GevIEEE1588Mode_SlaveOnly, [284](#)
- GevIEEE1588Status_Disabled, [284](#)
- GevIEEE1588Status_Faulty, [284](#)
- GevIEEE1588Status_Initializing, [284](#)
- GevIEEE1588Status_Listening, [285](#)
- GevIEEE1588Status_Master, [285](#)
- GevIEEE1588Status_Passive, [285](#)
- GevIEEE1588Status_PreMaster, [285](#)
- GevIEEE1588Status_Slave, [285](#)
- GevIEEE1588Status_Uncalibrated, [285](#)
- GevIPConfigurationStatus_DHCP, [285](#)
- GevIPConfigurationStatus_ForceIP, [285](#)
- GevIPConfigurationStatus_LLA, [285](#)
- GevIPConfigurationStatus_None, [285](#)
- GevIPConfigurationStatus_PersistentIP, [285](#)
- GevPhysicalLinkConfiguration_DynamicLAG, [285](#)
- GevPhysicalLinkConfiguration_MultiLink, [285](#)
- GevPhysicalLinkConfiguration_SingleLink, [285](#)
- GevPhysicalLinkConfiguration_StaticLAG, [285](#)
- GevSupportedOptionSelector_Action, [286](#)
- GevSupportedOptionSelector_CCPApplicationSocket, [286](#)
- GevSupportedOptionSelector_CommandsConcatenation, [286](#)
- GevSupportedOptionSelector_DiscoveryAckDelay, [286](#)
- GevSupportedOptionSelector_DiscoveryAckDelayWritable, [286](#)
- GevSupportedOptionSelector_Event, [286](#)
- GevSupportedOptionSelector_EventData, [286](#)
- GevSupportedOptionSelector_ExtendedStatusCodes, [286](#)
- GevSupportedOptionSelector_HeartbeatDisable, [286](#)
- GevSupportedOptionSelector_IPConfigurationDHCP, [286](#)
- GevSupportedOptionSelector_IPConfigurationLLA, [286](#)
- GevSupportedOptionSelector_IPConfigurationPersistentIP, [286](#)
- GevSupportedOptionSelector_LinkSpeed, [286](#)
- GevSupportedOptionSelector_ManifestTable, [286](#)
- GevSupportedOptionSelector_MessageChannelSourceSocket, [286](#)
- GevSupportedOptionSelector_PacketResend, [286](#)
- GevSupportedOptionSelector_PendingAck, [286](#)
- GevSupportedOptionSelector_SerialNumber, [286](#)
- GevSupportedOptionSelector_StreamChannelSourceSocket, [286](#)
- GevSupportedOptionSelector_TestData, [286](#)
- GevSupportedOptionSelector_UserDefinedName, [286](#)
- GevSupportedOptionSelector_WriteMem, [286](#)
- ImageComponentSelector_Color, [286](#)
- ImageComponentSelector_Confidence, [287](#)
- ImageComponentSelector_Disparity, [287](#)
- ImageComponentSelector_Infrared, [286](#)
- ImageComponentSelector_Intensity, [286](#)
- ImageComponentSelector_Range, [286](#)
- ImageComponentSelector_Scatter, [287](#)
- ImageComponentSelector_Ultraviolet, [286](#)
- ImageCompressionJPEGFormatOption_BaselineOptimized, [287](#)
- ImageCompressionJPEGFormatOption_BaselineStandard, [287](#)
- ImageCompressionJPEGFormatOption_Lossless, [287](#)
- ImageCompressionJPEGFormatOption_Progressive, [287](#)
- ImageCompressionMode_Lossless, [288](#)
- ImageCompressionMode_Off, [288](#)
- ImageCompressionRateOption_FixBitrate, [288](#)
- ImageCompressionRateOption_FixQuality, [288](#)
- LineFormat_LVDS, [288](#)
- LineFormat_NoConnect, [288](#)
- LineFormat_OpenDrain, [288](#)
- LineFormat_OptoCoupled, [288](#)
- LineFormat_RS422, [288](#)
- LineFormat_TriState, [288](#)
- LineFormat_TTL, [288](#)
- LineInputFilterSelector_Debounce, [289](#)
- LineInputFilterSelector_Deglitch, [289](#)
- LineMode_Input, [289](#)
- LineMode_Output, [289](#)

- LineSelector_Line0, [289](#)
- LineSelector_Line1, [289](#)
- LineSelector_Line2, [289](#)
- LineSelector_Line3, [289](#)
- LineSource_AllPixel, [290](#)
- LineSource_AnyPixel, [290](#)
- LineSource_Counter0Active, [290](#)
- LineSource_Counter1Active, [290](#)
- LineSource_ExposureActive, [290](#)
- LineSource_FrameTriggerWait, [290](#)
- LineSource_Line0, [290](#)
- LineSource_Line1, [290](#)
- LineSource_Line2, [290](#)
- LineSource_Line3, [290](#)
- LineSource_LogicBlock0, [290](#)
- LineSource_LogicBlock1, [290](#)
- LineSource_Off, [290](#)
- LineSource_PPSSignal, [290](#)
- LineSource_SerialPort0, [290](#)
- LineSource_UserOutput0, [290](#)
- LineSource_UserOutput1, [290](#)
- LineSource_UserOutput2, [290](#)
- LineSource_UserOutput3, [290](#)
- LogicBlockLUTInputActivation_AnyEdge, [290](#)
- LogicBlockLUTInputActivation_FallingEdge, [290](#)
- LogicBlockLUTInputActivation_LevelHigh, [290](#)
- LogicBlockLUTInputActivation_LevelLow, [290](#)
- LogicBlockLUTInputActivation_RisingEdge, [290](#)
- LogicBlockLUTInputSelector_Input0, [291](#)
- LogicBlockLUTInputSelector_Input1, [291](#)
- LogicBlockLUTInputSelector_Input2, [291](#)
- LogicBlockLUTInputSelector_Input3, [291](#)
- LogicBlockLUTInputSource_AcquisitionActive, [291](#)
- LogicBlockLUTInputSource_Counter0End, [291](#)
- LogicBlockLUTInputSource_Counter0Start, [291](#)
- LogicBlockLUTInputSource_Counter1End, [291](#)
- LogicBlockLUTInputSource_Counter1Start, [291](#)
- LogicBlockLUTInputSource_ExposureEnd, [291](#)
- LogicBlockLUTInputSource_ExposureStart, [291](#)
- LogicBlockLUTInputSource_FrameTriggerWait, [291](#)
- LogicBlockLUTInputSource_Line0, [291](#)
- LogicBlockLUTInputSource_Line1, [291](#)
- LogicBlockLUTInputSource_Line2, [291](#)
- LogicBlockLUTInputSource_Line3, [291](#)
- LogicBlockLUTInputSource_LogicBlock0, [291](#)
- LogicBlockLUTInputSource_LogicBlock1, [291](#)
- LogicBlockLUTInputSource_UserOutput0, [291](#)
- LogicBlockLUTInputSource_UserOutput1, [291](#)
- LogicBlockLUTInputSource_UserOutput2, [291](#)
- LogicBlockLUTInputSource_UserOutput3, [291](#)
- LogicBlockLUTInputSource_Zero, [291](#)
- LogicBlockLUTSelector_Enable, [292](#)
- LogicBlockLUTSelector_Value, [292](#)
- LogicBlockSelector_LogicBlock0, [292](#)
- LogicBlockSelector_LogicBlock1, [292](#)
- LUTSelector_LUT1, [292](#)
- NUM_ACQUISITIONMODE, [242](#)
- NUM_ACQUISITIONSTATUSSELECTOR, [242](#)
- NUM_ACTIONUNCONDITIONALMODE, [243](#)
- NUM_ADCBITDEPTH, [243](#)
- NUM_AUTOALGORITHMSELECTOR, [243](#)
- NUM_AUTOEXPOSURECONTROLPRIORITY, [244](#)
- NUM_AUTOEXPOSURELIGHTINGMODE, [244](#)
- NUM_AUTOEXPOSUREMETERINGMODE, [245](#)
- NUM_AUTOEXPOSURETARGETGREYVALUEAUTO, [245](#)
- NUM_BALANCERATIOSELECTOR, [245](#)
- NUM_BALANCEWHITEAUTO, [246](#)
- NUM_BALANCEWHITEAUTOPROFILE, [246](#)
- NUM_BINNINGHORIZONTALMODE, [246](#)
- NUM_BINNINGSELECTOR, [247](#)
- NUM_BINNINGVERTICALMODE, [247](#)
- NUM_BLACKLEVELAUTO, [248](#)
- NUM_BLACKLEVELAUTOBALANCE, [247](#)
- NUM_BLACKLEVELSELECTOR, [248](#)
- NUM_CHUNKBLACKLEVELSELECTOR, [248](#)
- NUM_CHUNKCOUNTERSELECTOR, [249](#)
- NUM_CHUNKENCODERSELECTOR, [249](#)
- NUM_CHUNKENCODERSTATUS, [249](#)
- NUM_CHUNKEXPOSURETIMESELECTOR, [250](#)
- NUM_CHUNKGAINSELECTOR, [250](#)
- NUM_CHUNKIMAGECOMPONENT, [251](#)
- NUM_CHUNKPIXELFORMAT, [251](#)
- NUM_CHUNKREGIONID, [251](#)
- NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR, [252](#)
- NUM_CHUNKSCAN3DCOORDINATESELECTOR, [252](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEM, [252](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE, [253](#)
- NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR, [253](#)
- NUM_CHUNKSCAN3DDISTANCEUNIT, [253](#)
- NUM_CHUNKSCAN3DOUTPUTMODE, [254](#)
- NUM_CHUNKSELECTOR, [255](#)
- NUM_CHUNKSOURCEID, [255](#)
- NUM_CHUNKTIMERSELECTOR, [256](#)
- NUM_CHUNKTRANSFERSTREAMID, [256](#)
- NUM_CLCONFIGURATION, [256](#)
- NUM_CLTIMESLOTSCOUNT, [257](#)
- NUM_COLORTRANSFORMATIONSELECTOR, [257](#)
- NUM_COLORTRANSFORMATIONVALUESELECTOR, [257](#)
- NUM_COMPRESSIONSATURATIONPRIORITY, [258](#)
- NUM_COUNTEREVENTACTIVATION, [258](#)
- NUM_COUNTEREVENTSOURCE, [259](#)
- NUM_COUNTERRESETACTIVATION, [259](#)
- NUM_COUNTERRESETSOURCE, [260](#)
- NUM_COUNTERSELECTOR, [260](#)
- NUM_COUNTERSTATUS, [260](#)

- NUM_COUNTERTRIGGERACTIVATION, [261](#)
- NUM_COUNTERTRIGGERSOURCE, [261](#)
- NUM_CXPCONNECTIONTESTMODE, [262](#)
- NUM_CXPLINKCONFIGURATION, [263](#)
- NUM_CXPLINKCONFIGURATIONPREFERRED, [264](#)
- NUM_CXPLINKCONFIGURATIONSTATUS, [265](#)
- NUM_CXPPOCXPSTATUS, [265](#)
- NUM_DECIMATIONHORIZONTALMODE, [265](#)
- NUM_DECIMATIONSELECTOR, [266](#)
- NUM_DECIMATIONVERTICALMODE, [266](#)
- NUM_DEFECTCORRECTIONMODE, [266](#)
- NUM_DEINTERLACING, [267](#)
- NUM_DEVICECHARACTERSET, [267](#)
- NUM_DEVICECLOCKSELECTOR, [267](#)
- NUM_DEVICECONNECTIONSTATUS, [267](#)
- NUM_DEVICEINDICATORMODE, [268](#)
- NUM_DEVICELINKHEARTBEATMODE, [268](#)
- NUM_DEVICELINKTHROUGHPUTLIMITMODE, [268](#)
- NUM_DEVICEPOWERSUPPLYSELECTOR, [269](#)
- NUM_DEVICEREGISTERSENDIANNNESS, [269](#)
- NUM_DEVICESCANTYPE, [269](#)
- NUM_DEVICESERIALPORTBAUDRATE, [270](#)
- NUM_DEVICESERIALPORTSELECTOR, [270](#)
- NUM_DEVICESTREAMCHANNELENDIANNNESS, [270](#)
- NUM_DEVICESTREAMCHANNELTYPE, [271](#)
- NUM_DEVICETAPGEOMETRY, [272](#)
- NUM_DEVICETEMPERATURESELECTOR, [272](#)
- NUM_DEVICETLTYPE, [272](#)
- NUM_DEVICETYPE, [273](#)
- NUM_ENCODERMODE, [273](#)
- NUM_ENCODEROUTPUTMODE, [274](#)
- NUM_ENCODERRESETACTIVATION, [274](#)
- NUM_ENCODERRESETSOURCE, [275](#)
- NUM_ENCODERSELECTOR, [275](#)
- NUM_ENCODERSOURCEA, [276](#)
- NUM_ENCODERSOURCEB, [276](#)
- NUM_ENCODERSTATUS, [276](#)
- NUM_EVENTNOTIFICATION, [277](#)
- NUM_EVENTSELECTOR, [277](#)
- NUM_EXPOSUREACTIVEMODE, [277](#)
- NUM_EXPOSUREAUTO, [278](#)
- NUM_EXPOSUREMODE, [278](#)
- NUM_EXPOSURETIMEMODE, [278](#)
- NUM_EXPOSURETIMESELECTOR, [279](#)
- NUM_FILEOPENMODE, [279](#)
- NUM_FILEOPERATIONSELECTOR, [280](#)
- NUM_FILEOPERATIONSTATUS, [280](#)
- NUM_FILESELECTOR, [280](#)
- NUM_GAINAUTO, [282](#)
- NUM_GAINAUTOBALANCE, [282](#)
- NUM_GAINSELECTOR, [282](#)
- NUM_GEVCCEP, [283](#)
- NUM_GEVCURRENTPHYSICALLINKCONFIGURATION, [283](#)
- NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR, [283](#)
- NUM_GEVGVSPEXTENDEDIDMODE, [284](#)
- NUM_GEVIIEEE1588CLOCKACCURACY, [284](#)
- NUM_GEVIIEEE1588MODE, [284](#)
- NUM_GEVIIEEE1588STATUS, [285](#)
- NUM_GEVIPCONFIGURATIONSTATUS, [285](#)
- NUM_GEVPHYSICALLINKCONFIGURATION, [285](#)
- NUM_GEVSUPPORTEDOPTIONSELECTOR, [286](#)
- NUM_IMAGECOMPONENTSELECTOR, [287](#)
- NUM_IMAGECOMPRESSIONJPEGFORMATOPTION, [287](#)
- NUM_IMAGECOMPRESSIONMODE, [288](#)
- NUM_IMAGECOMPRESSIONRATEOPTION, [288](#)
- NUM_LINEFORMAT, [288](#)
- NUM_LINEINPUTFILTERSELECTOR, [289](#)
- NUM_LINEMODE, [289](#)
- NUM_LINESELECTOR, [289](#)
- NUM_LINESOURCE, [290](#)
- NUM_LOGICBLOCKLUTINPUTACTIVATION, [290](#)
- NUM_LOGICBLOCKLUTINPUTSELECTOR, [291](#)
- NUM_LOGICBLOCKLUTINPUTSOURCE, [291](#)
- NUM_LOGICBLOCKLUTSELECTOR, [292](#)
- NUM_LOGICBLOCKSELECTOR, [292](#)
- NUM_LUTSELECTOR, [292](#)
- NUM_PIXELCOLORFILTER, [293](#)
- NUM_PIXELFORMAT, [298](#)
- NUM_PIXELFORMATINFOSELECTOR, [304](#)
- NUM_PIXELSIZE, [305](#)
- NUM_REGIONDESTINATION, [305](#)
- NUM_REGIONMODE, [305](#)
- NUM_REGIONSELECTOR, [306](#)
- NUM_RGBTRANSFORMLIGHTSOURCE, [306](#)
- NUM_SCAN3DCOORDINATEREFERENCESELECTOR, [307](#)
- NUM_SCAN3DCOORDINATESELECTOR, [307](#)
- NUM_SCAN3DCOORDINATESYSTEM, [307](#)
- NUM_SCAN3DCOORDINATESYSTEMREFERENCE, [308](#)
- NUM_SCAN3DCOORDINATETRANSFORMSELECTOR, [308](#)
- NUM_SCAN3DDISTANCEUNIT, [308](#)
- NUM_SCAN3DOUTPUTMODE, [309](#)
- NUM_SENSORDIGITIZATIONTAPS, [310](#)
- NUM_SENSORSHUTTERMODE, [310](#)
- NUM_SENSORTAPS, [310](#)
- NUM_SEQUENCERCONFIGURATIONMODE, [311](#)
- NUM_SEQUENCERCONFIGURATIONVALID, [311](#)
- NUM_SEQUENCERMODE, [311](#)
- NUM_SEQUENCERSETVALID, [312](#)
- NUM_SEQUENCERTRIGGERACTIVATION, [312](#)
- NUM_SEQUENCERTRIGGERSOURCE, [312](#)
- NUM_SERIALPORTBAUDRATE, [313](#)
- NUM_SERIALPORTPARITY, [313](#)
- NUM_SERIALPORTSELECTOR, [313](#)
- NUM_SERIALPORTSOURCE, [314](#)

NUM_SERIALPORTSTOPBITS, [314](#)
NUM_SOFTWARESIGNALSELECTOR, [314](#)
NUM_SOURCESELECTOR, [315](#)
NUM_TESTPATTERN, [315](#)
NUM_TESTPATTERNGENERATORSELECTOR, [315](#)
NUM_TIMERSELECTOR, [316](#)
NUM_TIMERSTATUS, [316](#)
NUM_TIMERTRIGGERACTIVATION, [316](#)
NUM_TIMERTRIGGERSOURCE, [318](#)
NUM_TRANSFERCOMPONENTSELECTOR, [318](#)
NUM_TRANSFERCONTROLMODE, [319](#)
NUM_TRANSFEROPERATIONMODE, [319](#)
NUM_TRANSFERQUEUEMODE, [319](#)
NUM_TRANSFERSELECTOR, [319](#)
NUM_TRANSFERSTATUSSELECTOR, [320](#)
NUM_TRANSFERTRIGGERACTIVATION, [320](#)
NUM_TRANSFERTRIGGERMODE, [321](#)
NUM_TRANSFERTRIGGERSELECTOR, [321](#)
NUM_TRANSFERTRIGGERSOURCE, [322](#)
NUM_TRIGGERACTIVATION, [322](#)
NUM_TRIGGERMODE, [323](#)
NUM_TRIGGEROVERLAP, [323](#)
NUM_TRIGGERSELECTOR, [323](#)
NUM_TRIGGERSOURCE, [324](#)
NUM_USEROUTPUTSELECTOR, [324](#)
NUM_USERSETDEFAULT, [325](#)
NUM_USERSETSELECTOR, [325](#)
NUM_WHITECLIPSELECTOR, [325](#)
PixelFormat_BayerBG, [293](#)
PixelFormat_BayerGB, [293](#)
PixelFormat_BayerGR, [293](#)
PixelFormat_BayerRG, [292](#)
PixelFormat_None, [292](#)
PixelFormat_B10, [295](#)
PixelFormat_B12, [295](#)
PixelFormat_B12_Jpeg, [298](#)
PixelFormat_B16, [295](#)
PixelFormat_B8, [295](#)
PixelFormat_BayerBG10, [294](#)
PixelFormat_BayerBG10p, [294](#)
PixelFormat_BayerBG10Packed, [294](#)
PixelFormat_BayerBG12, [294](#)
PixelFormat_BayerBG12p, [293](#)
PixelFormat_BayerBG12Packed, [293](#)
PixelFormat_BayerBG16, [293](#)
PixelFormat_BayerBG8, [293](#)
PixelFormat_BayerGB10, [294](#)
PixelFormat_BayerGB10p, [294](#)
PixelFormat_BayerGB10Packed, [294](#)
PixelFormat_BayerGB12, [294](#)
PixelFormat_BayerGB12p, [293](#)
PixelFormat_BayerGB12Packed, [293](#)
PixelFormat_BayerGB16, [293](#)
PixelFormat_BayerGB8, [293](#)
PixelFormat_BayerGR10, [294](#)
PixelFormat_BayerGR10p, [294](#)
PixelFormat_BayerGR10Packed, [293](#)
PixelFormat_BayerGR12, [294](#)
PixelFormat_BayerGR12p, [293](#)
PixelFormat_BayerGR12Packed, [293](#)
PixelFormat_BayerGR16, [293](#)
PixelFormat_BayerGR8, [293](#)
PixelFormat_BayerRGPolarized10p, [298](#)
PixelFormat_BayerRGPolarized12p, [298](#)
PixelFormat_BayerRGPolarized16, [298](#)
PixelFormat_BayerRGPolarized8, [298](#)
PixelFormat_BGR10, [295](#)
PixelFormat_BGR10p, [295](#)
PixelFormat_BGR12, [295](#)
PixelFormat_BGR12p, [295](#)
PixelFormat_BGR14, [295](#)
PixelFormat_BGR16, [295](#)
PixelFormat_BGR565p, [295](#)
PixelFormat_BGR8, [293](#)
PixelFormat_BGRa10, [295](#)
PixelFormat_BGRa10p, [295](#)
PixelFormat_BGRa12, [295](#)
PixelFormat_BGRa12p, [295](#)
PixelFormat_BGRa14, [295](#)
PixelFormat_BGRa16, [295](#)
PixelFormat_BGRa8, [293](#)
PixelFormat_BiColorBGRG10, [296](#)
PixelFormat_BiColorBGRG10p, [296](#)
PixelFormat_BiColorBGRG12, [296](#)
PixelFormat_BiColorBGRG12p, [296](#)
PixelFormat_BiColorBGRG8, [296](#)
PixelFormat_BiColorRGBG10, [296](#)
PixelFormat_BiColorRGBG10p, [296](#)
PixelFormat_BiColorRGBG12, [296](#)
PixelFormat_BiColorRGBG12p, [296](#)
PixelFormat_BiColorRGBG8, [296](#)
PixelFormat_Confidence1, [296](#)
PixelFormat_Confidence16, [296](#)
PixelFormat_Confidence1p, [296](#)
PixelFormat_Confidence32f, [296](#)
PixelFormat_Confidence8, [296](#)
PixelFormat_Coord3D_A10p, [296](#)
PixelFormat_Coord3D_A12p, [296](#)
PixelFormat_Coord3D_A16, [296](#)
PixelFormat_Coord3D_A32f, [296](#)
PixelFormat_Coord3D_A8, [296](#)
PixelFormat_Coord3D_ABC10p, [295](#)
PixelFormat_Coord3D_ABC10p_Planar, [295](#)
PixelFormat_Coord3D_ABC12p, [295](#)
PixelFormat_Coord3D_ABC12p_Planar, [295](#)
PixelFormat_Coord3D_ABC16, [295](#)
PixelFormat_Coord3D_ABC16_Planar, [295](#)

PixelFormat_Coord3D_ABC32f, 295
PixelFormat_Coord3D_ABC32f_Planar, 295
PixelFormat_Coord3D_ABC8, 295
PixelFormat_Coord3D_ABC8_Planar, 295
PixelFormat_Coord3D_AC10p, 295
PixelFormat_Coord3D_AC10p_Planar, 295
PixelFormat_Coord3D_AC12p, 295
PixelFormat_Coord3D_AC12p_Planar, 295
PixelFormat_Coord3D_AC16, 295
PixelFormat_Coord3D_AC16_Planar, 295
PixelFormat_Coord3D_AC32f, 295
PixelFormat_Coord3D_AC32f_Planar, 295
PixelFormat_Coord3D_AC8, 295
PixelFormat_Coord3D_AC8_Planar, 295
PixelFormat_Coord3D_B10p, 296
PixelFormat_Coord3D_B12p, 296
PixelFormat_Coord3D_B16, 296
PixelFormat_Coord3D_B32f, 296
PixelFormat_Coord3D_B8, 296
PixelFormat_Coord3D_C10p, 296
PixelFormat_Coord3D_C12p, 296
PixelFormat_Coord3D_C16, 296
PixelFormat_Coord3D_C32f, 296
PixelFormat_Coord3D_C8, 296
PixelFormat_G10, 295
PixelFormat_G12, 295
PixelFormat_G16, 295
PixelFormat_G8, 295
PixelFormat_GB12, 298
PixelFormat_GB12_Jpeg, 298
PixelFormat_GR12, 298
PixelFormat_GR12_Jpeg, 298
PixelFormat_JPEGColor8, 298
PixelFormat_JPEGMono8, 298
PixelFormat_LLCBayerRG8, 298
PixelFormat_LLCMono8, 298
PixelFormat_Mono10, 294
PixelFormat_Mono10p, 294
PixelFormat_Mono10Packed, 293
PixelFormat_Mono12, 294
PixelFormat_Mono12p, 293
PixelFormat_Mono12Packed, 293
PixelFormat_Mono14, 294
PixelFormat_Mono16, 293
PixelFormat_Mono16s, 294
PixelFormat_Mono1p, 294
PixelFormat_Mono2p, 294
PixelFormat_Mono32f, 294
PixelFormat_Mono4p, 294
PixelFormat_Mono8, 293
PixelFormat_Mono8s, 294
PixelFormat_Polarized10p, 298
PixelFormat_Polarized12p, 298
PixelFormat_Polarized16, 298
PixelFormat_Polarized8, 298
PixelFormat_R10, 295
PixelFormat_R12, 295
PixelFormat_R12_Jpeg, 298
PixelFormat_R16, 295
PixelFormat_R8, 295
PixelFormat_Raw16, 298
PixelFormat_Raw8, 298
PixelFormat_RGB10, 294
PixelFormat_RGB10_Planar, 294
PixelFormat_RGB10p, 294
PixelFormat_RGB10p32, 294
PixelFormat_RGB12, 294
PixelFormat_RGB12_Planar, 294
PixelFormat_RGB12p, 294
PixelFormat_RGB14, 294
PixelFormat_RGB16, 294
PixelFormat_RGB16_Planar, 294
PixelFormat_RGB16s, 294
PixelFormat_RGB32f, 294
PixelFormat_RGB565p, 295
PixelFormat_RGB8, 294
PixelFormat_RGB8_Planar, 294
PixelFormat_RGB8Packed, 293
PixelFormat_RGBa10, 294
PixelFormat_RGBa10p, 294
PixelFormat_RGBa12, 294
PixelFormat_RGBa12p, 294
PixelFormat_RGBa14, 294
PixelFormat_RGBa16, 294
PixelFormat_RGBa32f, 295
PixelFormat_RGBa8, 294
PixelFormat_SCF1WBWG10, 296
PixelFormat_SCF1WBWG10p, 296
PixelFormat_SCF1WBWG12, 296
PixelFormat_SCF1WBWG12p, 296
PixelFormat_SCF1WBWG14, 296
PixelFormat_SCF1WBWG16, 296
PixelFormat_SCF1WBWG8, 296
PixelFormat_SCF1WGWB10, 296
PixelFormat_SCF1WGWB10p, 296
PixelFormat_SCF1WGWB12, 296
PixelFormat_SCF1WGWB12p, 296
PixelFormat_SCF1WGWB14, 296
PixelFormat_SCF1WGWB16, 296
PixelFormat_SCF1WGWB8, 296
PixelFormat_SCF1WGWR10, 296
PixelFormat_SCF1WGWR10p, 296
PixelFormat_SCF1WGWR12, 297
PixelFormat_SCF1WGWR12p, 297
PixelFormat_SCF1WGWR14, 297
PixelFormat_SCF1WGWR16, 297
PixelFormat_SCF1WGWR8, 296
PixelFormat_SCF1WRWG10, 297
PixelFormat_SCF1WRWG10p, 297
PixelFormat_SCF1WRWG12, 297
PixelFormat_SCF1WRWG12p, 297
PixelFormat_SCF1WRWG14, 297
PixelFormat_SCF1WRWG16, 297
PixelFormat_SCF1WRWG8, 297
PixelFormat_YCbCr10_CbYCr, 297
PixelFormat_YCbCr10p_CbYCr, 297

- PixelFormat_YCbCr12_CbYCr, 297
- PixelFormat_YCbCr12p_CbYCr, 297
- PixelFormat_YCbCr411_8, 293
- PixelFormat_YCbCr411_8_CbYYCrYY, 297
- PixelFormat_YCbCr422_10, 297
- PixelFormat_YCbCr422_10_CbYCrY, 297
- PixelFormat_YCbCr422_10p, 297
- PixelFormat_YCbCr422_10p_CbYCrY, 297
- PixelFormat_YCbCr422_12, 297
- PixelFormat_YCbCr422_12_CbYCrY, 297
- PixelFormat_YCbCr422_12p, 297
- PixelFormat_YCbCr422_12p_CbYCrY, 297
- PixelFormat_YCbCr422_8, 293
- PixelFormat_YCbCr422_8_CbYCrY, 297
- PixelFormat_YCbCr601_10_CbYCr, 297
- PixelFormat_YCbCr601_10p_CbYCr, 297
- PixelFormat_YCbCr601_12_CbYCr, 297
- PixelFormat_YCbCr601_12p_CbYCr, 297
- PixelFormat_YCbCr601_411_8_CbYYCrYY, 297
- PixelFormat_YCbCr601_422_10, 297
- PixelFormat_YCbCr601_422_10_CbYCrY, 297
- PixelFormat_YCbCr601_422_10p, 297
- PixelFormat_YCbCr601_422_10p_CbYCrY, 297
- PixelFormat_YCbCr601_422_12, 297
- PixelFormat_YCbCr601_422_12_CbYCrY, 297
- PixelFormat_YCbCr601_422_12p, 297
- PixelFormat_YCbCr601_422_12p_CbYCrY, 297
- PixelFormat_YCbCr601_422_8, 297
- PixelFormat_YCbCr601_422_8_CbYCrY, 297
- PixelFormat_YCbCr601_8_CbYCr, 297
- PixelFormat_YCbCr709_10_CbYCr, 297
- PixelFormat_YCbCr709_10p_CbYCr, 297
- PixelFormat_YCbCr709_12_CbYCr, 297
- PixelFormat_YCbCr709_12p_CbYCr, 297
- PixelFormat_YCbCr709_411_8_CbYYCrYY, 298
- PixelFormat_YCbCr709_422_10, 298
- PixelFormat_YCbCr709_422_10_CbYCrY, 298
- PixelFormat_YCbCr709_422_10p, 298
- PixelFormat_YCbCr709_422_10p_CbYCrY, 298
- PixelFormat_YCbCr709_422_12, 298
- PixelFormat_YCbCr709_422_12_CbYCrY, 298
- PixelFormat_YCbCr709_422_12p, 298
- PixelFormat_YCbCr709_422_12p_CbYCrY, 298
- PixelFormat_YCbCr709_422_8, 298
- PixelFormat_YCbCr709_422_8_CbYCrY, 298
- PixelFormat_YCbCr709_8_CbYCr, 297
- PixelFormat_YCbCr8, 293
- PixelFormat_YCbCr8_CbYCr, 297
- PixelFormat_YUV411_8_UYVYY, 298
- PixelFormat_YUV411Packed, 293
- PixelFormat_YUV422_8, 298
- PixelFormat_YUV422_8_UYVY, 298
- PixelFormat_YUV422Packed, 293
- PixelFormat_YUV444Packed, 293
- PixelFormat_YUV8_UYV, 298
- PixelFormatInfoSelector_B10, 300
- PixelFormatInfoSelector_B12, 300
- PixelFormatInfoSelector_B16, 300
- PixelFormatInfoSelector_B8, 300
- PixelFormatInfoSelector_BayerBG10, 299
- PixelFormatInfoSelector_BayerBG10p, 299
- PixelFormatInfoSelector_BayerBG12, 299
- PixelFormatInfoSelector_BayerBG12p, 299
- PixelFormatInfoSelector_BayerBG16, 299
- PixelFormatInfoSelector_BayerBG8, 299
- PixelFormatInfoSelector_BayerGB10, 299
- PixelFormatInfoSelector_BayerGB10p, 299
- PixelFormatInfoSelector_BayerGB12, 299
- PixelFormatInfoSelector_BayerGB12p, 299
- PixelFormatInfoSelector_BayerGB16, 299
- PixelFormatInfoSelector_BayerGB8, 299
- PixelFormatInfoSelector_BayerGR10, 299
- PixelFormatInfoSelector_BayerGR10p, 299
- PixelFormatInfoSelector_BayerGR12, 299
- PixelFormatInfoSelector_BayerGR12p, 299
- PixelFormatInfoSelector_BayerGR16, 299
- PixelFormatInfoSelector_BayerGR8, 299
- PixelFormatInfoSelector_BayerRG10, 299
- PixelFormatInfoSelector_BayerRG10p, 299
- PixelFormatInfoSelector_BayerRG12, 299
- PixelFormatInfoSelector_BayerRG12p, 299
- PixelFormatInfoSelector_BayerRG16, 299
- PixelFormatInfoSelector_BayerRG8, 299
- PixelFormatInfoSelector_BayerRGPolarized10p, 304
- PixelFormatInfoSelector_BayerRGPolarized12p, 304
- PixelFormatInfoSelector_BayerRGPolarized16, 304
- PixelFormatInfoSelector_BayerRGPolarized8, 304
- PixelFormatInfoSelector_BGR10, 300
- PixelFormatInfoSelector_BGR10p, 300
- PixelFormatInfoSelector_BGR12, 300
- PixelFormatInfoSelector_BGR12p, 300
- PixelFormatInfoSelector_BGR14, 300
- PixelFormatInfoSelector_BGR16, 300
- PixelFormatInfoSelector_BGR565p, 300
- PixelFormatInfoSelector_BGR8, 300
- PixelFormatInfoSelector_BGRa10, 300
- PixelFormatInfoSelector_BGRa10p, 300
- PixelFormatInfoSelector_BGRa12, 300
- PixelFormatInfoSelector_BGRa12p, 300
- PixelFormatInfoSelector_BGRa14, 300
- PixelFormatInfoSelector_BGRa16, 300
- PixelFormatInfoSelector_BGRa8, 300
- PixelFormatInfoSelector_BiColorBGRG10, 301
- PixelFormatInfoSelector_BiColorBGRG10p, 301
- PixelFormatInfoSelector_BiColorBGRG12, 301
- PixelFormatInfoSelector_BiColorBGRG12p, 301
- PixelFormatInfoSelector_BiColorBGRG8, 301
- PixelFormatInfoSelector_BiColorRGBG10, 301
- PixelFormatInfoSelector_BiColorRGBG10p, 301
- PixelFormatInfoSelector_BiColorRGBG12, 301
- PixelFormatInfoSelector_BiColorRGBG12p, 301
- PixelFormatInfoSelector_BiColorRGBG8, 301
- PixelFormatInfoSelector_Confidence1, 301

- PixelFormatInfoSelector_Confidence16, [301](#)
- PixelFormatInfoSelector_Confidence1p, [301](#)
- PixelFormatInfoSelector_Confidence32f, [301](#)
- PixelFormatInfoSelector_Confidence8, [301](#)
- PixelFormatInfoSelector_Coord3D_A10p, [301](#)
- PixelFormatInfoSelector_Coord3D_A12p, [301](#)
- PixelFormatInfoSelector_Coord3D_A16, [301](#)
- PixelFormatInfoSelector_Coord3D_A32f, [301](#)
- PixelFormatInfoSelector_Coord3D_A8, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC10p, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC10p_Planar, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC12p, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC12p_Planar, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC16, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC16_Planar, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC32f, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC32f_Planar, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC8, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC8_Planar, [300](#)
- PixelFormatInfoSelector_Coord3D_AC10p, [301](#)
- PixelFormatInfoSelector_Coord3D_AC10p_Planar, [301](#)
- PixelFormatInfoSelector_Coord3D_AC12p, [301](#)
- PixelFormatInfoSelector_Coord3D_AC12p_Planar, [301](#)
- PixelFormatInfoSelector_Coord3D_AC16, [301](#)
- PixelFormatInfoSelector_Coord3D_AC16_Planar, [301](#)
- PixelFormatInfoSelector_Coord3D_AC32f, [301](#)
- PixelFormatInfoSelector_Coord3D_AC32f_Planar, [301](#)
- PixelFormatInfoSelector_Coord3D_AC8, [301](#)
- PixelFormatInfoSelector_Coord3D_AC8_Planar, [301](#)
- PixelFormatInfoSelector_Coord3D_B10p, [301](#)
- PixelFormatInfoSelector_Coord3D_B12p, [301](#)
- PixelFormatInfoSelector_Coord3D_B16, [301](#)
- PixelFormatInfoSelector_Coord3D_B32f, [301](#)
- PixelFormatInfoSelector_Coord3D_B8, [301](#)
- PixelFormatInfoSelector_Coord3D_C10p, [301](#)
- PixelFormatInfoSelector_Coord3D_C12p, [301](#)
- PixelFormatInfoSelector_Coord3D_C16, [301](#)
- PixelFormatInfoSelector_Coord3D_C32f, [301](#)
- PixelFormatInfoSelector_Coord3D_C8, [301](#)
- PixelFormatInfoSelector_G10, [300](#)
- PixelFormatInfoSelector_G12, [300](#)
- PixelFormatInfoSelector_G16, [300](#)
- PixelFormatInfoSelector_G8, [300](#)
- PixelFormatInfoSelector_JPEGColor8, [304](#)
- PixelFormatInfoSelector_JPEGMono8, [304](#)
- PixelFormatInfoSelector_LLCBayerRG8, [304](#)
- PixelFormatInfoSelector_LLCMono8, [304](#)
- PixelFormatInfoSelector_Mono10, [299](#)
- PixelFormatInfoSelector_Mono10p, [299](#)
- PixelFormatInfoSelector_Mono12, [299](#)
- PixelFormatInfoSelector_Mono12p, [299](#)
- PixelFormatInfoSelector_Mono14, [299](#)
- PixelFormatInfoSelector_Mono16, [299](#)
- PixelFormatInfoSelector_Mono16s, [299](#)
- PixelFormatInfoSelector_Mono1p, [299](#)
- PixelFormatInfoSelector_Mono2p, [299](#)
- PixelFormatInfoSelector_Mono32f, [299](#)
- PixelFormatInfoSelector_Mono4p, [299](#)
- PixelFormatInfoSelector_Mono8, [299](#)
- PixelFormatInfoSelector_Mono8s, [299](#)
- PixelFormatInfoSelector_Polarized10p, [304](#)
- PixelFormatInfoSelector_Polarized12p, [304](#)
- PixelFormatInfoSelector_Polarized16, [304](#)
- PixelFormatInfoSelector_Polarized8, [304](#)
- PixelFormatInfoSelector_R10, [300](#)
- PixelFormatInfoSelector_R12, [300](#)
- PixelFormatInfoSelector_R16, [300](#)
- PixelFormatInfoSelector_R8, [300](#)
- PixelFormatInfoSelector_RGB10, [299](#)
- PixelFormatInfoSelector_RGB10_Planar, [300](#)
- PixelFormatInfoSelector_RGB10p, [300](#)
- PixelFormatInfoSelector_RGB10p32, [300](#)
- PixelFormatInfoSelector_RGB12, [300](#)
- PixelFormatInfoSelector_RGB12_Planar, [300](#)
- PixelFormatInfoSelector_RGB12p, [300](#)
- PixelFormatInfoSelector_RGB14, [300](#)
- PixelFormatInfoSelector_RGB16, [300](#)
- PixelFormatInfoSelector_RGB16_Planar, [300](#)
- PixelFormatInfoSelector_RGB16s, [300](#)
- PixelFormatInfoSelector_RGB32f, [300](#)
- PixelFormatInfoSelector_RGB565p, [300](#)
- PixelFormatInfoSelector_RGB8, [299](#)
- PixelFormatInfoSelector_RGB8_Planar, [299](#)
- PixelFormatInfoSelector_RGBa10, [299](#)
- PixelFormatInfoSelector_RGBa10p, [299](#)
- PixelFormatInfoSelector_RGBa12, [299](#)
- PixelFormatInfoSelector_RGBa12p, [299](#)
- PixelFormatInfoSelector_RGBa14, [299](#)
- PixelFormatInfoSelector_RGBa16, [299](#)
- PixelFormatInfoSelector_RGBa32f, [300](#)
- PixelFormatInfoSelector_RGBa8, [299](#)
- PixelFormatInfoSelector_SCF1WBWG10, [301](#)
- PixelFormatInfoSelector_SCF1WBWG10p, [302](#)
- PixelFormatInfoSelector_SCF1WBWG12, [302](#)
- PixelFormatInfoSelector_SCF1WBWG14, [302](#)
- PixelFormatInfoSelector_SCF1WBWG16, [302](#)
- PixelFormatInfoSelector_SCF1WBWG8, [301](#)
- PixelFormatInfoSelector_SCF1WGWB10, [302](#)
- PixelFormatInfoSelector_SCF1WGWB10p, [302](#)
- PixelFormatInfoSelector_SCF1WGWB12, [302](#)
- PixelFormatInfoSelector_SCF1WGWB12p, [302](#)
- PixelFormatInfoSelector_SCF1WGWB14, [302](#)
- PixelFormatInfoSelector_SCF1WGWB16, [302](#)
- PixelFormatInfoSelector_SCF1WGWB8, [302](#)
- PixelFormatInfoSelector_SCF1WGWR10, [302](#)

- PixelFormatInfoSelector_SCF1WGWR10p, [302](#)
- PixelFormatInfoSelector_SCF1WGWR12, [302](#)
- PixelFormatInfoSelector_SCF1WGWR12p, [302](#)
- PixelFormatInfoSelector_SCF1WGWR14, [302](#)
- PixelFormatInfoSelector_SCF1WGWR16, [302](#)
- PixelFormatInfoSelector_SCF1WGWR8, [302](#)
- PixelFormatInfoSelector_SCF1WRWG10, [302](#)
- PixelFormatInfoSelector_SCF1WRWG10p, [302](#)
- PixelFormatInfoSelector_SCF1WRWG12, [302](#)
- PixelFormatInfoSelector_SCF1WRWG12p, [302](#)
- PixelFormatInfoSelector_SCF1WRWG14, [302](#)
- PixelFormatInfoSelector_SCF1WRWG16, [302](#)
- PixelFormatInfoSelector_SCF1WRWG8, [302](#)
- PixelFormatInfoSelector_YCbCr10_CbYCr, [302](#)
- PixelFormatInfoSelector_YCbCr10p_CbYCr, [302](#)
- PixelFormatInfoSelector_YCbCr12_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr12p_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr411_8, [303](#)
- PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY, [303](#)
- PixelFormatInfoSelector_YCbCr422_10, [303](#)
- PixelFormatInfoSelector_YCbCr422_10_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr422_10p, [303](#)
- PixelFormatInfoSelector_YCbCr422_10p_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr422_12, [303](#)
- PixelFormatInfoSelector_YCbCr422_12_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr422_12p, [303](#)
- PixelFormatInfoSelector_YCbCr422_12p_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr422_8, [303](#)
- PixelFormatInfoSelector_YCbCr422_8_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr601_10_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr601_10p_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr601_12_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr601_12p_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_10, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_10p, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_12, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_12p, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_8, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr601_422_8_CbYYCrYY, [303](#)
- PixelFormatInfoSelector_YCbCr601_8_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr601_8_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr709_10_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr709_10p_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr709_12_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr709_12p_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_10, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_10p, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_12, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_12p, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_8, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY, [303](#)
- PixelFormatInfoSelector_YCbCr709_8_CbYCr, [303](#)
- PixelFormatInfoSelector_YCbCr8, [302](#)
- PixelFormatInfoSelector_YCbCr8_CbYCr, [302](#)
- PixelFormatInfoSelector_YUV411_8_UYYVYY, [304](#)
- PixelFormatInfoSelector_YUV422_8, [304](#)
- PixelFormatInfoSelector_YUV422_8_UYVY, [304](#)
- PixelFormatInfoSelector_YUV8_UYV, [304](#)
- PixelSize_Bpp1, [304](#)
- PixelSize_Bpp10, [304](#)
- PixelSize_Bpp12, [304](#)
- PixelSize_Bpp14, [304](#)
- PixelSize_Bpp16, [304](#)
- PixelSize_Bpp2, [304](#)
- PixelSize_Bpp20, [304](#)
- PixelSize_Bpp24, [304](#)
- PixelSize_Bpp30, [304](#)
- PixelSize_Bpp32, [304](#)
- PixelSize_Bpp36, [305](#)
- PixelSize_Bpp4, [304](#)
- PixelSize_Bpp48, [305](#)
- PixelSize_Bpp64, [305](#)
- PixelSize_Bpp8, [304](#)
- PixelSize_Bpp96, [305](#)
- RegionDestination_Stream0, [305](#)
- RegionDestination_Stream1, [305](#)
- RegionDestination_Stream2, [305](#)
- RegionMode_Off, [305](#)

- RegionMode_On, 305
- RegionSelector_All, 306
- RegionSelector_Region0, 306
- RegionSelector_Region1, 306
- RegionSelector_Region2, 306
- RgbTransformLightSource_Cloudy6500K, 306
- RgbTransformLightSource_CoolFluorescent4000K, 306
- RgbTransformLightSource_Custom, 306
- RgbTransformLightSource_Daylight5000K, 306
- RgbTransformLightSource_General, 306
- RgbTransformLightSource_Shade8000K, 306
- RgbTransformLightSource_Tungsten2800K, 306
- RgbTransformLightSource_WarmFluorescent3000K, 306
- Scan3dCoordinateReferenceSelector_RotationX, 307
- Scan3dCoordinateReferenceSelector_RotationY, 307
- Scan3dCoordinateReferenceSelector_RotationZ, 307
- Scan3dCoordinateReferenceSelector_TranslationX, 307
- Scan3dCoordinateReferenceSelector_TranslationY, 307
- Scan3dCoordinateReferenceSelector_TranslationZ, 307
- Scan3dCoordinateSelector_CoordinateA, 307
- Scan3dCoordinateSelector_CoordinateB, 307
- Scan3dCoordinateSelector_CoordinateC, 307
- Scan3dCoordinateSystem_Cartesian, 307
- Scan3dCoordinateSystem_Cylindrical, 307
- Scan3dCoordinateSystem_Spherical, 307
- Scan3dCoordinateSystemReference_Anchor, 308
- Scan3dCoordinateSystemReference_Transformed, 308
- Scan3dCoordinateTransformSelector_RotationX, 308
- Scan3dCoordinateTransformSelector_RotationY, 308
- Scan3dCoordinateTransformSelector_RotationZ, 308
- Scan3dCoordinateTransformSelector_TranslationX, 308
- Scan3dCoordinateTransformSelector_TranslationY, 308
- Scan3dCoordinateTransformSelector_TranslationZ, 308
- Scan3dDistanceUnit_Inch, 308
- Scan3dDistanceUnit_Millimeter, 308
- Scan3dOutputMode_CalibratedABC_Grid, 309
- Scan3dOutputMode_CalibratedABC_PointCloud, 309
- Scan3dOutputMode_CalibratedAC, 309
- Scan3dOutputMode_CalibratedAC_Linescan, 309
- Scan3dOutputMode_CalibratedC, 309
- Scan3dOutputMode_CalibratedC_Linescan, 309
- Scan3dOutputMode_DisparityC, 309
- Scan3dOutputMode_DisparityC_Linescan, 309
- Scan3dOutputMode_RectifiedC, 309
- Scan3dOutputMode_RectifiedC_Linescan, 309
- Scan3dOutputMode_UncalibratedC, 309
- SensorDigitizationTaps_Eight, 310
- SensorDigitizationTaps_Four, 310
- SensorDigitizationTaps_One, 309
- SensorDigitizationTaps_Ten, 310
- SensorDigitizationTaps_Three, 310
- SensorDigitizationTaps_Two, 310
- SensorShutterMode_Global, 310
- SensorShutterMode_GlobalReset, 310
- SensorShutterMode_Rolling, 310
- SensorTaps_Eight, 310
- SensorTaps_Four, 310
- SensorTaps_One, 310
- SensorTaps_Ten, 310
- SensorTaps_Three, 310
- SensorTaps_Two, 310
- SequencerConfigurationMode_Off, 311
- SequencerConfigurationMode_On, 311
- SequencerConfigurationValid_No, 311
- SequencerConfigurationValid_Yes, 311
- SequencerMode_Off, 311
- SequencerMode_On, 311
- SequencerSetValid_No, 312
- SequencerSetValid_Yes, 312
- SequencerTriggerActivation_AnyEdge, 312
- SequencerTriggerActivation_FallingEdge, 312
- SequencerTriggerActivation_LevelHigh, 312
- SequencerTriggerActivation_LevelLow, 312
- SequencerTriggerActivation_RisingEdge, 312
- SequencerTriggerSource_FrameStart, 312
- SequencerTriggerSource_Off, 312
- SerialPortBaudRate_Baud115200, 313
- SerialPortBaudRate_Baud1200, 313
- SerialPortBaudRate_Baud14400, 313
- SerialPortBaudRate_Baud19200, 313
- SerialPortBaudRate_Baud230400, 313
- SerialPortBaudRate_Baud2400, 313
- SerialPortBaudRate_Baud300, 313
- SerialPortBaudRate_Baud38400, 313
- SerialPortBaudRate_Baud460800, 313
- SerialPortBaudRate_Baud4800, 313
- SerialPortBaudRate_Baud57600, 313
- SerialPortBaudRate_Baud600, 313
- SerialPortBaudRate_Baud921600, 313
- SerialPortBaudRate_Baud9600, 313
- SerialPortParity_Even, 313
- SerialPortParity_Mark, 313
- SerialPortParity_None, 313
- SerialPortParity_Odd, 313
- SerialPortParity_Space, 313
- SerialPortSelector_SerialPort0, 313
- SerialPortSource_Line0, 314
- SerialPortSource_Line1, 314
- SerialPortSource_Line2, 314
- SerialPortSource_Line3, 314

SerialPortSource_Off, 314
SerialPortStopBits_Bits1, 314
SerialPortStopBits_Bits1AndAHalf, 314
SerialPortStopBits_Bits2, 314
SoftwareSignalSelector_SoftwareSignal0, 314
SoftwareSignalSelector_SoftwareSignal1, 314
SoftwareSignalSelector_SoftwareSignal2, 314
SourceSelector_All, 315
SourceSelector_Source0, 315
SourceSelector_Source1, 315
SourceSelector_Source2, 315
TestPattern_Increment, 315
TestPattern_Off, 315
TestPattern_SensorTestPattern, 315
TestPatternGeneratorSelector_PipelineStart, 315
TestPatternGeneratorSelector_Sensor, 315
TimerSelector_Timer0, 316
TimerSelector_Timer1, 316
TimerSelector_Timer2, 316
TimerStatus_TimerActive, 316
TimerStatus_TimerCompleted, 316
TimerStatus_TimerIdle, 316
TimerStatus_TimerTriggerWait, 316
TimerTriggerActivation_AnyEdge, 316
TimerTriggerActivation_FallingEdge, 316
TimerTriggerActivation_LevelHigh, 316
TimerTriggerActivation_LevelLow, 316
TimerTriggerActivation_RisingEdge, 316
TimerTriggerSource_AcquisitionEnd, 317
TimerTriggerSource_AcquisitionStart, 317
TimerTriggerSource_AcquisitionTrigger, 317
TimerTriggerSource_Action0, 318
TimerTriggerSource_Action1, 318
TimerTriggerSource_Action2, 318
TimerTriggerSource_Counter0End, 317
TimerTriggerSource_Counter0Start, 317
TimerTriggerSource_Counter1End, 317
TimerTriggerSource_Counter1Start, 317
TimerTriggerSource_Counter2End, 317
TimerTriggerSource_Counter2Start, 317
TimerTriggerSource_Encoder0, 318
TimerTriggerSource_Encoder1, 318
TimerTriggerSource_Encoder2, 318
TimerTriggerSource_ExposureEnd, 317
TimerTriggerSource_ExposureStart, 317
TimerTriggerSource_FrameBurstEnd, 317
TimerTriggerSource_FrameBurstStart, 317
TimerTriggerSource_FrameEnd, 317
TimerTriggerSource_FrameStart, 317
TimerTriggerSource_FrameTrigger, 317
TimerTriggerSource_Line0, 317
TimerTriggerSource_Line1, 317
TimerTriggerSource_Line2, 317
TimerTriggerSource_LineEnd, 317
TimerTriggerSource_LineStart, 317
TimerTriggerSource_LineTrigger, 317
TimerTriggerSource_LinkTrigger0, 318
TimerTriggerSource_LinkTrigger1, 318
TimerTriggerSource_LinkTrigger2, 318
TimerTriggerSource_Off, 317
TimerTriggerSource_SoftwareSignal0, 318
TimerTriggerSource_SoftwareSignal1, 318
TimerTriggerSource_SoftwareSignal2, 318
TimerTriggerSource_Timer0End, 317
TimerTriggerSource_Timer0Start, 317
TimerTriggerSource_Timer1End, 317
TimerTriggerSource_Timer1Start, 317
TimerTriggerSource_Timer2End, 317
TimerTriggerSource_Timer2Start, 317
TimerTriggerSource_UserOutput0, 317
TimerTriggerSource_UserOutput1, 317
TimerTriggerSource_UserOutput2, 317
TransferComponentSelector_All, 318
TransferComponentSelector_Blue, 318
TransferComponentSelector_Green, 318
TransferComponentSelector_Red, 318
TransferControlMode_Automatic, 319
TransferControlMode_Basic, 319
TransferControlMode_UserControlled, 319
TransferOperationMode_Continuous, 319
TransferOperationMode_MultiBlock, 319
TransferQueueMode_FirstInFirstOut, 319
TransferSelector_All, 319
TransferSelector_Stream0, 319
TransferSelector_Stream1, 319
TransferSelector_Stream2, 319
TransferStatusSelector_Paused, 320
TransferStatusSelector_QueueOverflow, 320
TransferStatusSelector_Stopped, 320
TransferStatusSelector_Stopping, 320
TransferStatusSelector_Streaming, 320
TransferTriggerActivation_AnyEdge, 320
TransferTriggerActivation_FallingEdge, 320
TransferTriggerActivation_LevelHigh, 320
TransferTriggerActivation_LevelLow, 320
TransferTriggerActivation_RisingEdge, 320
TransferTriggerMode_Off, 321
TransferTriggerMode_On, 321
TransferTriggerSelector_TransferAbort, 321
TransferTriggerSelector_TransferActive, 321
TransferTriggerSelector_TransferBurstStart, 321
TransferTriggerSelector_TransferBurstStop, 321
TransferTriggerSelector_TransferPause, 321
TransferTriggerSelector_TransferResume, 321
TransferTriggerSelector_TransferStart, 321
TransferTriggerSelector_TransferStop, 321
TransferTriggerSource_Action0, 322
TransferTriggerSource_Action1, 322
TransferTriggerSource_Action2, 322
TransferTriggerSource_Counter0End, 322
TransferTriggerSource_Counter0Start, 321
TransferTriggerSource_Counter1End, 322
TransferTriggerSource_Counter1Start, 321
TransferTriggerSource_Counter2End, 322
TransferTriggerSource_Counter2Start, 322
TransferTriggerSource_Line0, 321

- TransferTriggerSource_Line1, [321](#)
- TransferTriggerSource_Line2, [321](#)
- TransferTriggerSource_SoftwareSignal0, [322](#)
- TransferTriggerSource_SoftwareSignal1, [322](#)
- TransferTriggerSource_SoftwareSignal2, [322](#)
- TransferTriggerSource_Timer0End, [322](#)
- TransferTriggerSource_Timer0Start, [322](#)
- TransferTriggerSource_Timer1End, [322](#)
- TransferTriggerSource_Timer1Start, [322](#)
- TransferTriggerSource_Timer2End, [322](#)
- TransferTriggerSource_Timer2Start, [322](#)
- TriggerActivation_AnyEdge, [322](#)
- TriggerActivation_FallingEdge, [322](#)
- TriggerActivation_LevelHigh, [322](#)
- TriggerActivation_LevelLow, [322](#)
- TriggerActivation_RisingEdge, [322](#)
- TriggerMode_Off, [323](#)
- TriggerMode_On, [323](#)
- TriggerOverlap_Off, [323](#)
- TriggerOverlap_PreviousFrame, [323](#)
- TriggerOverlap_ReadOut, [323](#)
- TriggerSelector_AcquisitionStart, [323](#)
- TriggerSelector_FrameBurstStart, [323](#)
- TriggerSelector_FrameStart, [323](#)
- TriggerSource_Action0, [324](#)
- TriggerSource_Counter0End, [324](#)
- TriggerSource_Counter0Start, [324](#)
- TriggerSource_Counter1End, [324](#)
- TriggerSource_Counter1Start, [324](#)
- TriggerSource_Line0, [324](#)
- TriggerSource_Line1, [324](#)
- TriggerSource_Line2, [324](#)
- TriggerSource_Line3, [324](#)
- TriggerSource_LogicBlock0, [324](#)
- TriggerSource_LogicBlock1, [324](#)
- TriggerSource_Software, [324](#)
- TriggerSource_UserOutput0, [324](#)
- TriggerSource_UserOutput1, [324](#)
- TriggerSource_UserOutput2, [324](#)
- TriggerSource_UserOutput3, [324](#)
- UNKNOWN_PIXELFORMAT, [298](#)
- UserOutputSelector_UserOutput0, [324](#)
- UserOutputSelector_UserOutput1, [324](#)
- UserOutputSelector_UserOutput2, [324](#)
- UserOutputSelector_UserOutput3, [324](#)
- UserSetDefault_Default, [325](#)
- UserSetDefault_UserSet0, [325](#)
- UserSetDefault_UserSet1, [325](#)
- UserSetSelector_Default, [325](#)
- UserSetSelector_UserSet0, [325](#)
- UserSetSelector_UserSet1, [325](#)
- WhiteClipSelector_All, [325](#)
- WhiteClipSelector_Blue, [325](#)
- WhiteClipSelector_Green, [325](#)
- WhiteClipSelector_Red, [325](#)
- WhiteClipSelector_Tap1, [325](#)
- WhiteClipSelector_Tap2, [325](#)
- WhiteClipSelector_U, [325](#)
- WhiteClipSelector_V, [325](#)
- WhiteClipSelector_Y, [325](#)
- CameraList Access, [30](#)
- CategoryNode
 - SpinnakerGenApiDefsC.h, [501](#)
- Chunk data access, [42](#)
- Chunk Data Structures, [23](#)
- ChunkBlackLevel
 - _quickSpin, [89](#)
- ChunkBlackLevelSelector
 - _quickSpin, [90](#)
- ChunkBlackLevelSelector_All
 - CameraDefsC.h, [248](#)
- ChunkCompressionMode
 - _quickSpin, [90](#)
- ChunkCompressionRatio
 - _quickSpin, [90](#)
- ChunkCounterSelector
 - _quickSpin, [90](#)
- ChunkCounterSelector_Counter0
 - CameraDefsC.h, [249](#)
- ChunkCounterSelector_Counter1
 - CameraDefsC.h, [249](#)
- ChunkCounterSelector_Counter2
 - CameraDefsC.h, [249](#)
- ChunkCounterValue
 - _quickSpin, [90](#)
- ChunkCRC
 - _quickSpin, [90](#)
- ChunkEnable
 - _quickSpin, [90](#)
- ChunkEncoderSelector
 - _quickSpin, [90](#)
- ChunkEncoderSelector_Encoder0
 - CameraDefsC.h, [249](#)
- ChunkEncoderSelector_Encoder1
 - CameraDefsC.h, [249](#)
- ChunkEncoderSelector_Encoder2
 - CameraDefsC.h, [249](#)
- ChunkEncoderStatus
 - _quickSpin, [91](#)
- ChunkEncoderStatus_EncoderDown
 - CameraDefsC.h, [249](#)
- ChunkEncoderStatus_EncoderIdle
 - CameraDefsC.h, [249](#)
- ChunkEncoderStatus_EncoderStatic
 - CameraDefsC.h, [249](#)
- ChunkEncoderStatus_EncoderUp
 - CameraDefsC.h, [249](#)
- ChunkEncoderValue
 - _quickSpin, [91](#)
- ChunkExposureEndLineStatusAll
 - _quickSpin, [91](#)
- ChunkExposureTime
 - _quickSpin, [91](#)
- ChunkExposureTimeSelector
 - _quickSpin, [91](#)
- ChunkExposureTimeSelector_Blue

- CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Common
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Cyan
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Green
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Infrared
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Magenta
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Red
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Stage1
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Stage2
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Ultraviolet
 - CameraDefsC.h, [250](#)
- ChunkExposureTimeSelector_Yellow
 - CameraDefsC.h, [250](#)
- ChunkFrameID
 - _quickSpin, [91](#)
- ChunkGain
 - _quickSpin, [91](#)
- ChunkGainSelector
 - _quickSpin, [91](#)
- ChunkGainSelector_All
 - CameraDefsC.h, [250](#)
- ChunkGainSelector_Blue
 - CameraDefsC.h, [250](#)
- ChunkGainSelector_Green
 - CameraDefsC.h, [250](#)
- ChunkGainSelector_Red
 - CameraDefsC.h, [250](#)
- ChunkHeight
 - _quickSpin, [92](#)
- ChunkImage
 - _quickSpin, [92](#)
- ChunkImageComponent
 - _quickSpin, [92](#)
- ChunkImageComponent_Color
 - CameraDefsC.h, [250](#)
- ChunkImageComponent_Confidence
 - CameraDefsC.h, [251](#)
- ChunkImageComponent_Disparity
 - CameraDefsC.h, [250](#)
- ChunkImageComponent_Infrared
 - CameraDefsC.h, [250](#)
- ChunkImageComponent_Intensity
 - CameraDefsC.h, [250](#)
- ChunkImageComponent_Range
 - CameraDefsC.h, [250](#)
- ChunkImageComponent_Scatter
 - CameraDefsC.h, [251](#)
- ChunkImageComponent_Ultraviolet
 - CameraDefsC.h, [250](#)
- ChunkInferenceBoundingBoxResult
 - _quickSpin, [92](#)
- ChunkInferenceConfidence
 - _quickSpin, [92](#)
- ChunkInferenceFrameId
 - _quickSpin, [92](#)
- ChunkInferenceResult
 - _quickSpin, [92](#)
- ChunkLinePitch
 - _quickSpin, [92](#)
- ChunkLineStatusAll
 - _quickSpin, [93](#)
- ChunkModeActive
 - _quickSpin, [93](#)
- ChunkOffsetX
 - _quickSpin, [93](#)
- ChunkOffsetY
 - _quickSpin, [93](#)
- ChunkPartSelector
 - _quickSpin, [93](#)
- ChunkPixelDynamicRangeMax
 - _quickSpin, [93](#)
- ChunkPixelDynamicRangeMin
 - _quickSpin, [93](#)
- ChunkPixelFormat
 - _quickSpin, [93](#)
- ChunkPixelFormat_BayerBG8
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_BayerGB8
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_BayerGR8
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_BayerRG8
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_Mono12Packed
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_Mono16
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_Mono8
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_RGB8Packed
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_YCbCr601_422_8_CbYCrY
 - CameraDefsC.h, [251](#)
- ChunkPixelFormat_YUV422Packed
 - CameraDefsC.h, [251](#)
- ChunkRegionID
 - _quickSpin, [94](#)
- ChunkRegionID_Region0
 - CameraDefsC.h, [251](#)
- ChunkRegionID_Region1
 - CameraDefsC.h, [251](#)
- ChunkRegionID_Region2
 - CameraDefsC.h, [251](#)
- ChunkScan3dAxisMax
 - _quickSpin, [94](#)
- ChunkScan3dAxisMin
 - _quickSpin, [94](#)
- ChunkScan3dCoordinateOffset

- [_quickSpin](#), 94
- ChunkScan3dCoordinateReferenceSelector
 - [_quickSpin](#), 94
- ChunkScan3dCoordinateReferenceSelector_RotationX
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateReferenceSelector_RotationY
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateReferenceSelector_RotationZ
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateReferenceSelector_TranslationX
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateReferenceSelector_TranslationY
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateReferenceSelector_TranslationZ
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateReferenceValue
 - [_quickSpin](#), 94
- ChunkScan3dCoordinateScale
 - [_quickSpin](#), 94
- ChunkScan3dCoordinateSelector
 - [_quickSpin](#), 94
- ChunkScan3dCoordinateSelector_CoordinateA
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateSelector_CoordinateB
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateSelector_CoordinateC
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateSystem
 - [_quickSpin](#), 95
- ChunkScan3dCoordinateSystem_Cartesian
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateSystem_Cylindrical
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateSystem_Spherical
 - CameraDefsC.h, 252
- ChunkScan3dCoordinateSystemReference
 - [_quickSpin](#), 95
- ChunkScan3dCoordinateSystemReference_Anchor
 - CameraDefsC.h, 253
- ChunkScan3dCoordinateSystemReference_Transformed
 - CameraDefsC.h, 253
- ChunkScan3dCoordinateTransformSelector
 - [_quickSpin](#), 95
- ChunkScan3dCoordinateTransformSelector_RotationX
 - CameraDefsC.h, 253
- ChunkScan3dCoordinateTransformSelector_RotationY
 - CameraDefsC.h, 253
- ChunkScan3dCoordinateTransformSelector_RotationZ
 - CameraDefsC.h, 253
- ChunkScan3dCoordinateTransformSelector_TranslationX
 - CameraDefsC.h, 253
- ChunkScan3dCoordinateTransformSelector_TranslationY
 - CameraDefsC.h, 253
- ChunkScan3dCoordinateTransformSelector_TranslationZ
 - CameraDefsC.h, 253
- ChunkScan3dDistanceUnit
 - [_quickSpin](#), 95
- ChunkScan3dDistanceUnit_Inch
 - CameraDefsC.h, 253
- ChunkScan3dDistanceUnit_Millimeter
 - CameraDefsC.h, 253
- ChunkScan3dInvalidDataFlag
 - [_quickSpin](#), 95
- ChunkScan3dInvalidDataValue
 - [_quickSpin](#), 95
- ChunkScan3dOutputMode
 - [_quickSpin](#), 95
- ChunkScan3dOutputMode_CalibratedABC_Grid
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_CalibratedABC_PointCloud
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_CalibratedAC
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_CalibratedAC_Linescan
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_CalibratedC
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_CalibratedC_Linescan
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_DisparityC
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_DisparityC_Linescan
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_RectifiedC
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_RectifiedC_Linescan
 - CameraDefsC.h, 254
- ChunkScan3dOutputMode_UncalibratedC
 - CameraDefsC.h, 254
- ChunkScan3dTransformValue
 - [_quickSpin](#), 95
- ChunkScanLineSelector
 - [_quickSpin](#), 96
- ChunkSelector
 - [_quickSpin](#), 96
- ChunkSelector_BlackLevel
 - CameraDefsC.h, 255
- ChunkSelector_CRC
 - CameraDefsC.h, 255
- ChunkSelector_ExposureEndLineStatusAll
 - CameraDefsC.h, 255
- ChunkSelector_ExposureTime
 - CameraDefsC.h, 255
- ChunkSelector_FrameID
 - CameraDefsC.h, 255
- ChunkSelector_Gain
 - CameraDefsC.h, 255
- ChunkSelector_Height
 - CameraDefsC.h, 255
- ChunkSelector_Image
 - CameraDefsC.h, 255
- ChunkSelector_OffsetX
 - CameraDefsC.h, 255
- ChunkSelector_OffsetY
 - CameraDefsC.h, 255
- ChunkSelector_PixelFormat

- CameraDefsC.h, [255](#)
- ChunkSelector_SequencerSetActive
 - CameraDefsC.h, [255](#)
- ChunkSelector_SerialData
 - CameraDefsC.h, [255](#)
- ChunkSelector_Timestamp
 - CameraDefsC.h, [255](#)
- ChunkSelector_Width
 - CameraDefsC.h, [255](#)
- ChunkSequencerSetActive
 - _quickSpin, [96](#)
- ChunkSerialData
 - _quickSpin, [96](#)
- ChunkSerialDataLength
 - _quickSpin, [96](#)
- ChunkSerialReceiveOverflow
 - _quickSpin, [96](#)
- ChunkSourceID
 - _quickSpin, [96](#)
- ChunkSourceID_Source0
 - CameraDefsC.h, [255](#)
- ChunkSourceID_Source1
 - CameraDefsC.h, [255](#)
- ChunkSourceID_Source2
 - CameraDefsC.h, [255](#)
- ChunkStreamChannelID
 - _quickSpin, [96](#)
- ChunkTimerSelector
 - _quickSpin, [97](#)
- ChunkTimerSelector_Timer0
 - CameraDefsC.h, [256](#)
- ChunkTimerSelector_Timer1
 - CameraDefsC.h, [256](#)
- ChunkTimerSelector_Timer2
 - CameraDefsC.h, [256](#)
- ChunkTimerValue
 - _quickSpin, [97](#)
- ChunkTimestamp
 - _quickSpin, [97](#)
- ChunkTimestampLatchValue
 - _quickSpin, [97](#)
- ChunkTransferBlockID
 - _quickSpin, [97](#)
- ChunkTransferQueueCurrentBlockCount
 - _quickSpin, [97](#)
- ChunkTransferStreamID
 - _quickSpin, [97](#)
- ChunkTransferStreamID_Stream0
 - CameraDefsC.h, [256](#)
- ChunkTransferStreamID_Stream1
 - CameraDefsC.h, [256](#)
- ChunkTransferStreamID_Stream2
 - CameraDefsC.h, [256](#)
- ChunkTransferStreamID_Stream3
 - CameraDefsC.h, [256](#)
- ChunkWidth
 - _quickSpin, [97](#)
- CL
- SpinnakerGenApiDefsC.h, [502](#)
- CIConfiguration
 - _quickSpin, [98](#)
- CIConfiguration_Base
 - CameraDefsC.h, [256](#)
- CIConfiguration_DualBase
 - CameraDefsC.h, [256](#)
- CIConfiguration_EightyBit
 - CameraDefsC.h, [256](#)
- CIConfiguration_Full
 - CameraDefsC.h, [256](#)
- CIConfiguration_Medium
 - CameraDefsC.h, [256](#)
- CITimeSlotsCount
 - _quickSpin, [98](#)
- CITimeSlotsCount_One
 - CameraDefsC.h, [257](#)
- CITimeSlotsCount_Three
 - CameraDefsC.h, [257](#)
- CITimeSlotsCount_Two
 - CameraDefsC.h, [257](#)
- ColorTransformationEnable
 - _quickSpin, [98](#)
- ColorTransformationSelector
 - _quickSpin, [98](#)
- ColorTransformationSelector_RGBtoRGB
 - CameraDefsC.h, [257](#)
- ColorTransformationSelector_RGBtoYUV
 - CameraDefsC.h, [257](#)
- ColorTransformationValue
 - _quickSpin, [98](#)
- ColorTransformationValueSelector
 - _quickSpin, [98](#)
- ColorTransformationValueSelector_Gain00
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Gain01
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Gain02
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Gain10
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Gain11
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Gain12
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Gain20
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Gain21
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Gain22
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Offset0
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Offset1
 - CameraDefsC.h, [257](#)
- ColorTransformationValueSelector_Offset2
 - CameraDefsC.h, [257](#)
- CommandNode

- SpinnakerGenApiDefsC.h, [500](#)
- compression
 - _spinTIFFOption, [207](#)
- compressionLevel
 - _spinPNGOption, [205](#)
- CompressionRatio
 - _quickSpin, [98](#)
- CompressionSaturationPriority
 - _quickSpin, [98](#)
- CompressionSaturationPriority_DropFrame
 - CameraDefsC.h, [258](#)
- CompressionSaturationPriority_ReduceFrameRate
 - CameraDefsC.h, [258](#)
- CounterDelay
 - _quickSpin, [99](#)
- CounterDuration
 - _quickSpin, [99](#)
- CounterEventActivation
 - _quickSpin, [99](#)
- CounterEventActivation_AnyEdge
 - CameraDefsC.h, [258](#)
- CounterEventActivation_FallingEdge
 - CameraDefsC.h, [258](#)
- CounterEventActivation_LevelHigh
 - CameraDefsC.h, [258](#)
- CounterEventActivation_LevelLow
 - CameraDefsC.h, [258](#)
- CounterEventActivation_RisingEdge
 - CameraDefsC.h, [258](#)
- CounterEventSource
 - _quickSpin, [99](#)
- CounterEventSource_Counter0End
 - CameraDefsC.h, [259](#)
- CounterEventSource_Counter0Start
 - CameraDefsC.h, [259](#)
- CounterEventSource_Counter1End
 - CameraDefsC.h, [259](#)
- CounterEventSource_Counter1Start
 - CameraDefsC.h, [259](#)
- CounterEventSource_ExposureEnd
 - CameraDefsC.h, [259](#)
- CounterEventSource_ExposureStart
 - CameraDefsC.h, [259](#)
- CounterEventSource_FrameTriggerWait
 - CameraDefsC.h, [259](#)
- CounterEventSource_Line0
 - CameraDefsC.h, [258](#)
- CounterEventSource_Line1
 - CameraDefsC.h, [259](#)
- CounterEventSource_Line2
 - CameraDefsC.h, [259](#)
- CounterEventSource_Line3
 - CameraDefsC.h, [259](#)
- CounterEventSource_LogicBlock0
 - CameraDefsC.h, [259](#)
- CounterEventSource_LogicBlock1
 - CameraDefsC.h, [259](#)
- CounterEventSource_MHzTick
 - CameraDefsC.h, [258](#)
- CounterEventSource_Off
 - CameraDefsC.h, [258](#)
- CounterEventSource_UserOutput0
 - CameraDefsC.h, [259](#)
- CounterEventSource_UserOutput1
 - CameraDefsC.h, [259](#)
- CounterEventSource_UserOutput2
 - CameraDefsC.h, [259](#)
- CounterEventSource_UserOutput3
 - CameraDefsC.h, [259](#)
- CounterReset
 - _quickSpin, [99](#)
- CounterResetActivation
 - _quickSpin, [99](#)
- CounterResetActivation_AnyEdge
 - CameraDefsC.h, [259](#)
- CounterResetActivation_FallingEdge
 - CameraDefsC.h, [259](#)
- CounterResetActivation_LevelHigh
 - CameraDefsC.h, [259](#)
- CounterResetActivation_LevelLow
 - CameraDefsC.h, [259](#)
- CounterResetActivation_RisingEdge
 - CameraDefsC.h, [259](#)
- CounterResetSource
 - _quickSpin, [99](#)
- CounterResetSource_Counter0End
 - CameraDefsC.h, [260](#)
- CounterResetSource_Counter0Start
 - CameraDefsC.h, [260](#)
- CounterResetSource_Counter1End
 - CameraDefsC.h, [260](#)
- CounterResetSource_Counter1Start
 - CameraDefsC.h, [260](#)
- CounterResetSource_ExposureEnd
 - CameraDefsC.h, [260](#)
- CounterResetSource_ExposureStart
 - CameraDefsC.h, [260](#)
- CounterResetSource_FrameTriggerWait
 - CameraDefsC.h, [260](#)
- CounterResetSource_Line0
 - CameraDefsC.h, [260](#)
- CounterResetSource_Line1
 - CameraDefsC.h, [260](#)
- CounterResetSource_Line2
 - CameraDefsC.h, [260](#)
- CounterResetSource_Line3
 - CameraDefsC.h, [260](#)
- CounterResetSource_LogicBlock0
 - CameraDefsC.h, [260](#)
- CounterResetSource_LogicBlock1
 - CameraDefsC.h, [260](#)
- CounterResetSource_Off
 - CameraDefsC.h, [259](#)
- CounterResetSource_UserOutput0
 - CameraDefsC.h, [260](#)
- CounterResetSource_UserOutput1
 - CameraDefsC.h, [260](#)

- CameraDefsC.h, [260](#)
- CounterResetSource_UserOutput2
 - CameraDefsC.h, [260](#)
- CounterResetSource_UserOutput3
 - CameraDefsC.h, [260](#)
- CounterSelector
 - _quickSpin, [99](#)
- CounterSelector_Counter0
 - CameraDefsC.h, [260](#)
- CounterSelector_Counter1
 - CameraDefsC.h, [260](#)
- CounterStatus
 - _quickSpin, [100](#)
- CounterStatus_CounterActive
 - CameraDefsC.h, [260](#)
- CounterStatus_CounterCompleted
 - CameraDefsC.h, [260](#)
- CounterStatus_CounterIdle
 - CameraDefsC.h, [260](#)
- CounterStatus_CounterOverflow
 - CameraDefsC.h, [260](#)
- CounterStatus_CounterTriggerWait
 - CameraDefsC.h, [260](#)
- CounterTriggerActivation
 - _quickSpin, [100](#)
- CounterTriggerActivation_AnyEdge
 - CameraDefsC.h, [261](#)
- CounterTriggerActivation_FallingEdge
 - CameraDefsC.h, [261](#)
- CounterTriggerActivation_LevelHigh
 - CameraDefsC.h, [261](#)
- CounterTriggerActivation_LevelLow
 - CameraDefsC.h, [261](#)
- CounterTriggerActivation_RisingEdge
 - CameraDefsC.h, [261](#)
- CounterTriggerSource
 - _quickSpin, [100](#)
- CounterTriggerSource_Counter0End
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_Counter0Start
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_Counter1End
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_Counter1Start
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_ExposureEnd
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_ExposureStart
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_FrameTriggerWait
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_Line0
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_Line1
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_Line2
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_Line3
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_LogicBlock0
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_LogicBlock1
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_Off
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_UserOutput0
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_UserOutput1
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_UserOutput2
 - CameraDefsC.h, [261](#)
- CounterTriggerSource_UserOutput3
 - CameraDefsC.h, [261](#)
- CounterValue
 - _quickSpin, [100](#)
- CounterValueAtReset
 - _quickSpin, [100](#)
- ctAllDependingNodes
 - SpinnakerGenApiDefsC.h, [500](#)
- ctAllTerminalNodes
 - SpinnakerGenApiDefsC.h, [500](#)
- ctDependingChildren
 - SpinnakerGenApiDefsC.h, [500](#)
- ctInvalidators
 - SpinnakerGenApiDefsC.h, [500](#)
- ctReadingChildren
 - SpinnakerGenApiDefsC.h, [500](#)
- ctWritingChildren
 - SpinnakerGenApiDefsC.h, [500](#)
- Custom
 - SpinnakerGenApiDefsC.h, [500](#)
- CxpConnectionSelector
 - _quickSpin, [100](#)
- CxpConnectionTestErrorCount
 - _quickSpin, [100](#)
- CxpConnectionTestMode
 - _quickSpin, [100](#)
- CxpConnectionTestMode_Mode1
 - CameraDefsC.h, [262](#)
- CxpConnectionTestMode_Off
 - CameraDefsC.h, [262](#)
- CxpConnectionTestPacketCount
 - _quickSpin, [101](#)
- CxpLinkConfiguration
 - _quickSpin, [101](#)
- CxpLinkConfiguration_Auto
 - CameraDefsC.h, [262](#)
- CxpLinkConfiguration_CXP1_X1
 - CameraDefsC.h, [262](#)
- CxpLinkConfiguration_CXP1_X2
 - CameraDefsC.h, [262](#)
- CxpLinkConfiguration_CXP1_X3
 - CameraDefsC.h, [262](#)
- CxpLinkConfiguration_CXP1_X4
 - CameraDefsC.h, [262](#)
- CxpLinkConfiguration_CXP1_X5

CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP1_X6
 CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP2_X1
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP2_X2
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP2_X3
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP2_X4
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP2_X5
 CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP2_X6
 CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP3_X1
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP3_X2
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP3_X3
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP3_X4
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP3_X5
 CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP3_X6
 CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP5_X1
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP5_X2
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP5_X3
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP5_X4
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP5_X5
 CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP5_X6
 CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP6_X1
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP6_X2
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP6_X3
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP6_X4
 CameraDefsC.h, [262](#)
 CxpLinkConfiguration_CXP6_X5
 CameraDefsC.h, [263](#)
 CxpLinkConfiguration_CXP6_X6
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred
 _quickSpin, [101](#)
 CxpLinkConfigurationPreferred_CXP1_X1
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP1_X2
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP1_X3

CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP1_X4
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP1_X5
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP1_X6
 CameraDefsC.h, [264](#)
 CxpLinkConfigurationPreferred_CXP2_X1
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP2_X2
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP2_X3
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP2_X4
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP2_X5
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP2_X6
 CameraDefsC.h, [264](#)
 CxpLinkConfigurationPreferred_CXP3_X1
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP3_X2
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP3_X3
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP3_X4
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP3_X5
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP3_X6
 CameraDefsC.h, [264](#)
 CxpLinkConfigurationPreferred_CXP5_X1
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP5_X2
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP5_X3
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP5_X4
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP5_X5
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP5_X6
 CameraDefsC.h, [264](#)
 CxpLinkConfigurationPreferred_CXP6_X1
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP6_X2
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP6_X3
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP6_X4
 CameraDefsC.h, [263](#)
 CxpLinkConfigurationPreferred_CXP6_X5
 CameraDefsC.h, [264](#)
 CxpLinkConfigurationPreferred_CXP6_X6
 CameraDefsC.h, [264](#)
 CxpLinkConfigurationStatus
 _quickSpin, [101](#)
 CxpLinkConfigurationStatus_CXP1_X1

- CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP1_X2
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP1_X3
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP1_X4
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP1_X5
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP1_X6
 - CameraDefsC.h, [265](#)
- CxpLinkConfigurationStatus_CXP2_X1
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP2_X2
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP2_X3
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP2_X4
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP2_X5
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP2_X6
 - CameraDefsC.h, [265](#)
- CxpLinkConfigurationStatus_CXP3_X1
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP3_X2
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP3_X3
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP3_X4
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP3_X5
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP3_X6
 - CameraDefsC.h, [265](#)
- CxpLinkConfigurationStatus_CXP5_X1
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP5_X2
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP5_X3
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP5_X4
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP5_X5
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP5_X6
 - CameraDefsC.h, [265](#)
- CxpLinkConfigurationStatus_CXP6_X1
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP6_X2
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP6_X3
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP6_X4
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_CXP6_X5
 - CameraDefsC.h, [265](#)
- CxpLinkConfigurationStatus_CXP6_X6
 - CameraDefsC.h, [265](#)
- CameraDefsC.h, [265](#)
- CxpLinkConfigurationStatus_None
 - CameraDefsC.h, [264](#)
- CxpLinkConfigurationStatus_Pending
 - CameraDefsC.h, [264](#)
- CxpPoCxpAuto
 - _quickSpin, [101](#)
- CxpPoCxpStatus
 - _quickSpin, [101](#)
- CxpPoCxpStatus_Auto
 - CameraDefsC.h, [265](#)
- CxpPoCxpStatus_Off
 - CameraDefsC.h, [265](#)
- CxpPoCxpStatus_Tripped
 - CameraDefsC.h, [265](#)
- CxpPoCxpTripReset
 - _quickSpin, [101](#)
- CxpPoCxpTurnOff
 - _quickSpin, [101](#)
- DecimationHorizontal
 - _quickSpin, [102](#)
- DecimationHorizontalMode
 - _quickSpin, [102](#)
- DecimationHorizontalMode_Discard
 - CameraDefsC.h, [265](#)
- DecimationSelector
 - _quickSpin, [102](#)
- DecimationSelector_All
 - CameraDefsC.h, [266](#)
- DecimationSelector_Sensor
 - CameraDefsC.h, [266](#)
- DecimationVertical
 - _quickSpin, [102](#)
- DecimationVerticalMode
 - _quickSpin, [102](#)
- DecimationVerticalMode_Discard
 - CameraDefsC.h, [266](#)
- Decreasing
 - SpinnakerGenApiDefsC.h, [502](#)
- DefectCorrectionMode
 - _quickSpin, [102](#)
- DefectCorrectionMode_Average
 - CameraDefsC.h, [266](#)
- DefectCorrectionMode_Highlight
 - CameraDefsC.h, [266](#)
- DefectCorrectionMode_Zero
 - CameraDefsC.h, [266](#)
- DefectCorrectStaticEnable
 - _quickSpin, [102](#)
- DefectTableApply
 - _quickSpin, [102](#)
- DefectTableCoordinateX
 - _quickSpin, [103](#)
- DefectTableCoordinateY
 - _quickSpin, [103](#)
- DefectTableFactoryRestore
 - _quickSpin, [103](#)
- DefectTableIndex

- [_quickSpin, 103](#)
- DefectTablePixelCount
 - [_quickSpin, 103](#)
- DefectTableSave
 - [_quickSpin, 103](#)
- Deinterlacing
 - [_quickSpin, 103](#)
- Deinterlacing_LineDuplication
 - [CameraDefsC.h, 267](#)
- Deinterlacing_Off
 - [CameraDefsC.h, 267](#)
- Deinterlacing_Weave
 - [CameraDefsC.h, 267](#)
- Device Event Data Access, [41](#)
- DeviceAccessStatus
 - [_quickSpinTLDevice, 167](#)
 - [_quickSpinTLInterface, 174](#)
- DeviceAccessStatus_Busy
 - [TransportLayerDefsC.h, 509](#)
- DeviceAccessStatus_NoAccess
 - [TransportLayerDefsC.h, 509](#)
- DeviceAccessStatus_OpenReadOnly
 - [TransportLayerDefsC.h, 509](#)
- DeviceAccessStatus_OpenReadWrite
 - [TransportLayerDefsC.h, 509](#)
- DeviceAccessStatus_ReadOnly
 - [TransportLayerDefsC.h, 509](#)
- DeviceAccessStatus_ReadWrite
 - [TransportLayerDefsC.h, 509](#)
- DeviceAccessStatus_Unknown
 - [TransportLayerDefsC.h, 509](#)
- DeviceAddress
 - [_actionCommandResult, 69](#)
- DeviceBootloaderVersion
 - [_quickSpinTLDevice, 168](#)
- DeviceCharacterSet
 - [_quickSpin, 103](#)
- DeviceCharacterSet_ASCII
 - [CameraDefsC.h, 267](#)
- DeviceCharacterSet_UTF8
 - [CameraDefsC.h, 267](#)
- DeviceClockFrequency
 - [_quickSpin, 104](#)
- DeviceClockSelector
 - [_quickSpin, 104](#)
- DeviceClockSelector_CameraLink
 - [CameraDefsC.h, 267](#)
- DeviceClockSelector_Sensor
 - [CameraDefsC.h, 267](#)
- DeviceClockSelector_SensorDigitization
 - [CameraDefsC.h, 267](#)
- DeviceConnectionSelector
 - [_quickSpin, 104](#)
- DeviceConnectionSpeed
 - [_quickSpin, 104](#)
- DeviceConnectionStatus
 - [_quickSpin, 104](#)
- DeviceConnectionStatus_Active
 - [CameraDefsC.h, 267](#)
- DeviceConnectionStatus_Inactive
 - [CameraDefsC.h, 267](#)
- DeviceCount
 - [_quickSpinTLInterface, 174](#)
- DeviceCurrentSpeed
 - [_quickSpinTLDevice, 168](#)
- DeviceCurrentSpeed_FullSpeed
 - [TransportLayerDefsC.h, 509](#)
- DeviceCurrentSpeed_HighSpeed
 - [TransportLayerDefsC.h, 509](#)
- DeviceCurrentSpeed_LowSpeed
 - [TransportLayerDefsC.h, 509](#)
- DeviceCurrentSpeed_SuperSpeed
 - [TransportLayerDefsC.h, 509](#)
- DeviceCurrentSpeed_UnknownSpeed
 - [TransportLayerDefsC.h, 509](#)
- DeviceDisplayName
 - [_quickSpinTLDevice, 168](#)
- DeviceDriverVersion
 - [_quickSpinTLDevice, 168](#)
- DeviceEndiannessMechanism
 - [_quickSpinTLDevice, 168](#)
- DeviceEndiannessMechanism_Legacy
 - [TransportLayerDefsC.h, 511](#)
- DeviceEndiannessMechanism_Standard
 - [TransportLayerDefsC.h, 511](#)
- DeviceEventChannelCount
 - [_quickSpin, 104](#)
- DeviceFamilyName
 - [_quickSpin, 104](#)
- DeviceFeaturePersistenceEnd
 - [_quickSpin, 104](#)
- DeviceFeaturePersistenceStart
 - [_quickSpin, 105](#)
- DeviceFirmwareVersion
 - [_quickSpin, 105](#)
- DeviceGenCPVersionMajor
 - [_quickSpin, 105](#)
- DeviceGenCPVersionMinor
 - [_quickSpin, 105](#)
- DeviceID
 - [_quickSpin, 105](#)
 - [_quickSpinTLDevice, 168](#)
 - [_quickSpinTLInterface, 175](#)
- DeviceIndicatorMode
 - [_quickSpin, 105](#)
- DeviceIndicatorMode_Active
 - [CameraDefsC.h, 268](#)
- DeviceIndicatorMode_ErrorStatus
 - [CameraDefsC.h, 268](#)
- DeviceIndicatorMode_Inactive
 - [CameraDefsC.h, 268](#)
- DeviceInstancelId
 - [_quickSpinTLDevice, 168](#)
- DeviceIsUpdater
 - [_quickSpinTLDevice, 168](#)
- DeviceLinkBandwidthReserve

- [_quickSpin](#), 105
- DeviceLinkCommandTimeout
 - [_quickSpin](#), 105
- DeviceLinkConnectionCount
 - [_quickSpin](#), 106
- DeviceLinkCurrentThroughput
 - [_quickSpin](#), 106
- DeviceLinkHeartbeatMode
 - [_quickSpin](#), 106
- DeviceLinkHeartbeatMode_Off
 - CameraDefsC.h, 268
- DeviceLinkHeartbeatMode_On
 - CameraDefsC.h, 268
- DeviceLinkHeartbeatTimeout
 - [_quickSpin](#), 106
- DeviceLinkSelector
 - [_quickSpin](#), 106
- DeviceLinkSpeed
 - [_quickSpin](#), 106
 - [_quickSpinTLDevice](#), 169
- DeviceLinkThroughputLimit
 - [_quickSpin](#), 106
- DeviceLinkThroughputLimitMode
 - [_quickSpin](#), 106
- DeviceLinkThroughputLimitMode_Off
 - CameraDefsC.h, 268
- DeviceLinkThroughputLimitMode_On
 - CameraDefsC.h, 268
- DeviceLocation
 - [_quickSpinTLDevice](#), 169
- DeviceManifestEntrySelector
 - [_quickSpin](#), 107
- DeviceManifestPrimaryURL
 - [_quickSpin](#), 107
- DeviceManifestSchemaMajorVersion
 - [_quickSpin](#), 107
- DeviceManifestSchemaMinorVersion
 - [_quickSpin](#), 107
- DeviceManifestSecondaryURL
 - [_quickSpin](#), 107
- DeviceManifestXMLMajorVersion
 - [_quickSpin](#), 107
- DeviceManifestXMLMinorVersion
 - [_quickSpin](#), 107
- DeviceManifestXMLSubMinorVersion
 - [_quickSpin](#), 107
- DeviceManufacturerInfo
 - [_quickSpin](#), 108
- DeviceMaxThroughput
 - [_quickSpin](#), 108
- DeviceModelName
 - [_quickSpin](#), 108
 - [_quickSpinTLDevice](#), 169
 - [_quickSpinTLInterface](#), 175
- DeviceMulticastMonitorMode
 - [_quickSpinTLDevice](#), 169
- DevicePortId
 - [_quickSpinTLDevice](#), 169
- DevicePowerSupplySelector
 - [_quickSpin](#), 108
- DevicePowerSupplySelector_External
 - CameraDefsC.h, 269
- DeviceRegistersCheck
 - [_quickSpin](#), 108
- DeviceRegistersEndianness
 - [_quickSpin](#), 108
- DeviceRegistersEndianness_Big
 - CameraDefsC.h, 269
- DeviceRegistersEndianness_Little
 - CameraDefsC.h, 269
- DeviceRegistersStreamingEnd
 - [_quickSpin](#), 108
- DeviceRegistersStreamingStart
 - [_quickSpin](#), 108
- DeviceRegistersValid
 - [_quickSpin](#), 109
- DeviceReset
 - [_quickSpin](#), 109
 - [_quickSpinTLDevice](#), 169
- DeviceScanType
 - [_quickSpin](#), 109
- DeviceScanType_Areascan
 - CameraDefsC.h, 269
- DeviceSelector
 - [_quickSpinTLInterface](#), 175
- DeviceSerialNumber
 - [_quickSpin](#), 109
 - [_quickSpinTLDevice](#), 169
 - [_quickSpinTLInterface](#), 175
- DeviceSerialPortBaudRate
 - [_quickSpin](#), 109
- DeviceSerialPortBaudRate_Baud115200
 - CameraDefsC.h, 270
- DeviceSerialPortBaudRate_Baud19200
 - CameraDefsC.h, 270
- DeviceSerialPortBaudRate_Baud230400
 - CameraDefsC.h, 270
- DeviceSerialPortBaudRate_Baud38400
 - CameraDefsC.h, 270
- DeviceSerialPortBaudRate_Baud460800
 - CameraDefsC.h, 270
- DeviceSerialPortBaudRate_Baud57600
 - CameraDefsC.h, 270
- DeviceSerialPortBaudRate_Baud921600
 - CameraDefsC.h, 270
- DeviceSerialPortBaudRate_Baud9600
 - CameraDefsC.h, 270
- DeviceSerialPortSelector
 - [_quickSpin](#), 109
- DeviceSerialPortSelector_CameraLink
 - CameraDefsC.h, 270
- DeviceSFNCVersionMajor
 - [_quickSpin](#), 109
- DeviceSFNCVersionMinor
 - [_quickSpin](#), 109
- DeviceSFNCVersionSubMinor

- [_quickSpin](#), 110
- DeviceStreamChannelCount
 - [_quickSpin](#), 110
- DeviceStreamChannelEndianness
 - [_quickSpin](#), 110
- DeviceStreamChannelEndianness_Big
 - [CameraDefsC.h](#), 270
- DeviceStreamChannelEndianness_Little
 - [CameraDefsC.h](#), 270
- DeviceStreamChannelLink
 - [_quickSpin](#), 110
- DeviceStreamChannelPacketSize
 - [_quickSpin](#), 110
- DeviceStreamChannelSelector
 - [_quickSpin](#), 110
- DeviceStreamChannelType
 - [_quickSpin](#), 110
- DeviceStreamChannelType_Receiver
 - [CameraDefsC.h](#), 271
- DeviceStreamChannelType_Transmitter
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry
 - [_quickSpin](#), 110
- DeviceTapGeometry_Geometry_10X
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_10X_1Y
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_1X
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X10
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_1X10_1Y
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_1X2
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X2_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X2_1Y2
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X2_2YE
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X3
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X3_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X4
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X4_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X8
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_1X8_1Y
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_1X_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X_1Y2
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_1X_2YE
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2X
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2X2
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2X2_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2X2E
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2X2E_2YE
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_2X2M
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_2X_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2X_2YE
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2XE
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2XE_1Y2
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2XE_2YE
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2XM
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2XM_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2XM_1Y2
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_2XM_2YE
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_3X
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_3X_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_4X
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_4X2
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_4X2_1Y
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_4X2E
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_4X2E_1Y
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_4X_1Y
 - [CameraDefsC.h](#), 271
- DeviceTapGeometry_Geometry_8X
 - [CameraDefsC.h](#), 272
- DeviceTapGeometry_Geometry_8X_1Y
 - [CameraDefsC.h](#), 272
- DeviceTemperature
 - [_quickSpin](#), 111
- DeviceTemperatureSelector

- [_quickSpin](#), 111
- DeviceTemperatureSelector_Sensor
 - [CameraDefsC.h](#), 272
- DeviceTLType
 - [_quickSpin](#), 111
- DeviceTLType_CameraLink
 - [CameraDefsC.h](#), 272
- DeviceTLType_CameraLinkHS
 - [CameraDefsC.h](#), 272
- DeviceTLType_CoaXPress
 - [CameraDefsC.h](#), 272
- DeviceTLType_Custom
 - [CameraDefsC.h](#), 272
- DeviceTLType_GigEVision
 - [CameraDefsC.h](#), 272
- DeviceTLType_USB3Vision
 - [CameraDefsC.h](#), 272
- DeviceTLVersionMajor
 - [_quickSpin](#), 111
- DeviceTLVersionMinor
 - [_quickSpin](#), 111
- DeviceTLVersionSubMinor
 - [_quickSpin](#), 111
- DeviceType
 - [_quickSpin](#), 111
 - [_quickSpinTLDevice](#), 169
- DeviceType_CameraLink
 - [TransportLayerDefsC.h](#), 511
- DeviceType_CameraLinkHS
 - [TransportLayerDefsC.h](#), 511
- DeviceType_CoaXPress
 - [TransportLayerDefsC.h](#), 511
- DeviceType_Custom
 - [TransportLayerDefsC.h](#), 511
- DeviceType_GigEVision
 - [TransportLayerDefsC.h](#), 511
- DeviceType_Peripheral
 - [CameraDefsC.h](#), 273
- DeviceType_Receiver
 - [CameraDefsC.h](#), 273
- DeviceType_Transceiver
 - [CameraDefsC.h](#), 273
- DeviceType_Transmitter
 - [CameraDefsC.h](#), 273
- DeviceType_USB3Vision
 - [TransportLayerDefsC.h](#), 511
- DeviceU3VProtocol
 - [_quickSpinTLDevice](#), 170
- DeviceUnlock
 - [_quickSpinTLInterface](#), 175
- DeviceUpdateList
 - [_quickSpinTLInterface](#), 175
- DeviceUptime
 - [_quickSpin](#), 111
- DeviceUserID
 - [_quickSpin](#), 112
 - [_quickSpinTLDevice](#), 170
- DeviceVendorName
 - [_quickSpin](#), 112
 - [_quickSpinTLDevice](#), 170
 - [_quickSpinTLInterface](#), 175
- DeviceVersion
 - [_quickSpin](#), 112
 - [_quickSpinTLDevice](#), 170
- [doc/spindocs/C/GettingStarted.dox](#), 209
- [doc/spindocs/C/ProgrammerGuide.dox](#), 209
- [doc/spindocs/shared/Benefits.dox](#), 209
- [doc/spindocs/shared/FlyCapture2Comparison.dox](#), 209
- [doc/spindocs/shared/GenICamGenTL.dox](#), 209
- [doc/spindocs/shared/Licensing.dox](#), 209
- [doc/spindocs/shared/Maintenance.dox](#), 209
- EncoderDivider
 - [_quickSpin](#), 112
- EncoderMode
 - [_quickSpin](#), 112
- EncoderMode_FourPhase
 - [CameraDefsC.h](#), 273
- EncoderMode_HighResolution
 - [CameraDefsC.h](#), 273
- EncoderOutputMode
 - [_quickSpin](#), 112
- EncoderOutputMode_DirectionDown
 - [CameraDefsC.h](#), 274
- EncoderOutputMode_DirectionUp
 - [CameraDefsC.h](#), 274
- EncoderOutputMode_Motion
 - [CameraDefsC.h](#), 274
- EncoderOutputMode_Off
 - [CameraDefsC.h](#), 273
- EncoderOutputMode_PositionDown
 - [CameraDefsC.h](#), 274
- EncoderOutputMode_PositionUp
 - [CameraDefsC.h](#), 273
- EncoderReset
 - [_quickSpin](#), 112
- EncoderResetActivation
 - [_quickSpin](#), 112
- EncoderResetActivation_AnyEdge
 - [CameraDefsC.h](#), 274
- EncoderResetActivation_FallingEdge
 - [CameraDefsC.h](#), 274
- EncoderResetActivation_LevelHigh
 - [CameraDefsC.h](#), 274
- EncoderResetActivation_LevelLow
 - [CameraDefsC.h](#), 274
- EncoderResetActivation_RisingEdge
 - [CameraDefsC.h](#), 274
- EncoderResetSource
 - [_quickSpin](#), 113
- EncoderResetSource_AcquisitionEnd
 - [CameraDefsC.h](#), 274
- EncoderResetSource_AcquisitionStart
 - [CameraDefsC.h](#), 274
- EncoderResetSource_AcquisitionTrigger
 - [CameraDefsC.h](#), 274
- EncoderResetSource_Action0

- CameraDefsC.h, [275](#)
- EncoderResetSource_Action1
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Action2
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Counter0End
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Counter0Start
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Counter1End
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Counter1Start
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Counter2End
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Counter2Start
 - CameraDefsC.h, [275](#)
- EncoderResetSource_ExposureEnd
 - CameraDefsC.h, [275](#)
- EncoderResetSource_ExposureStart
 - CameraDefsC.h, [274](#)
- EncoderResetSource_FrameEnd
 - CameraDefsC.h, [274](#)
- EncoderResetSource_FrameStart
 - CameraDefsC.h, [274](#)
- EncoderResetSource_FrameTrigger
 - CameraDefsC.h, [274](#)
- EncoderResetSource_Line0
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Line1
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Line2
 - CameraDefsC.h, [275](#)
- EncoderResetSource_LinkTrigger0
 - CameraDefsC.h, [275](#)
- EncoderResetSource_LinkTrigger1
 - CameraDefsC.h, [275](#)
- EncoderResetSource_LinkTrigger2
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Off
 - CameraDefsC.h, [274](#)
- EncoderResetSource_SoftwareSignal0
 - CameraDefsC.h, [275](#)
- EncoderResetSource_SoftwareSignal1
 - CameraDefsC.h, [275](#)
- EncoderResetSource_SoftwareSignal2
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Timer0End
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Timer0Start
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Timer1End
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Timer1Start
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Timer2End
 - CameraDefsC.h, [275](#)
- EncoderResetSource_Timer2Start
 - CameraDefsC.h, [275](#)
- EncoderResetSource_UserOutput0
 - CameraDefsC.h, [275](#)
- EncoderResetSource_UserOutput1
 - CameraDefsC.h, [275](#)
- EncoderResetSource_UserOutput2
 - CameraDefsC.h, [275](#)
- EncoderSelector
 - _quickSpin, [113](#)
- EncoderSelector_Encoder0
 - CameraDefsC.h, [275](#)
- EncoderSelector_Encoder1
 - CameraDefsC.h, [275](#)
- EncoderSelector_Encoder2
 - CameraDefsC.h, [275](#)
- EncoderSourceA
 - _quickSpin, [113](#)
- EncoderSourceA_Line0
 - CameraDefsC.h, [276](#)
- EncoderSourceA_Line1
 - CameraDefsC.h, [276](#)
- EncoderSourceA_Line2
 - CameraDefsC.h, [276](#)
- EncoderSourceA_Off
 - CameraDefsC.h, [276](#)
- EncoderSourceB
 - _quickSpin, [113](#)
- EncoderSourceB_Line0
 - CameraDefsC.h, [276](#)
- EncoderSourceB_Line1
 - CameraDefsC.h, [276](#)
- EncoderSourceB_Line2
 - CameraDefsC.h, [276](#)
- EncoderSourceB_Off
 - CameraDefsC.h, [276](#)
- EncoderStatus
 - _quickSpin, [113](#)
- EncoderStatus_EncoderDown
 - CameraDefsC.h, [276](#)
- EncoderStatus_EncoderIdle
 - CameraDefsC.h, [276](#)
- EncoderStatus_EncoderStatic
 - CameraDefsC.h, [276](#)
- EncoderStatus_EncoderUp
 - CameraDefsC.h, [276](#)
- EncoderTimeout
 - _quickSpin, [113](#)
- EncoderValue
 - _quickSpin, [113](#)
- EncoderValueAtReset
 - _quickSpin, [113](#)
- EnumEntryNode
 - SpinnakerGenApiDefsC.h, [501](#)
- EnumerateGen2Cameras
 - _quickSpinTLSystem, [186](#)
- EnumerateGEVInterfaces
 - _quickSpinTLSystem, [186](#)
- EnumerateUSBInterfaces

- [_quickSpinTLSystem, 186](#)
- EnumerationCount
 - [_quickSpin, 114](#)
- EnumerationNode
 - [SpinnakerGenApiDefsC.h, 501](#)
- Error Handling, [27](#)
- Event Access, [38](#)
- EventAcquisitionEnd
 - [_quickSpin, 114](#)
- EventAcquisitionEndFrameID
 - [_quickSpin, 114](#)
- EventAcquisitionEndTimestamp
 - [_quickSpin, 114](#)
- EventAcquisitionError
 - [_quickSpin, 114](#)
- EventAcquisitionErrorFrameID
 - [_quickSpin, 114](#)
- EventAcquisitionErrorTimestamp
 - [_quickSpin, 114](#)
- EventAcquisitionStart
 - [_quickSpin, 114](#)
- EventAcquisitionStartFrameID
 - [_quickSpin, 115](#)
- EventAcquisitionStartTimestamp
 - [_quickSpin, 115](#)
- EventAcquisitionTransferEnd
 - [_quickSpin, 115](#)
- EventAcquisitionTransferEndFrameID
 - [_quickSpin, 115](#)
- EventAcquisitionTransferEndTimestamp
 - [_quickSpin, 115](#)
- EventAcquisitionTransferStart
 - [_quickSpin, 115](#)
- EventAcquisitionTransferStartFrameID
 - [_quickSpin, 115](#)
- EventAcquisitionTransferStartTimestamp
 - [_quickSpin, 115](#)
- EventAcquisitionTrigger
 - [_quickSpin, 116](#)
- EventAcquisitionTriggerFrameID
 - [_quickSpin, 116](#)
- EventAcquisitionTriggerTimestamp
 - [_quickSpin, 116](#)
- EventActionLate
 - [_quickSpin, 116](#)
- EventActionLateFrameID
 - [_quickSpin, 116](#)
- EventActionLateTimestamp
 - [_quickSpin, 116](#)
- EventCounter0End
 - [_quickSpin, 116](#)
- EventCounter0EndFrameID
 - [_quickSpin, 116](#)
- EventCounter0EndTimestamp
 - [_quickSpin, 117](#)
- EventCounter0Start
 - [_quickSpin, 117](#)
- EventCounter0StartFrameID
 - [_quickSpin, 117](#)
- EventCounter0StartTimestamp
 - [_quickSpin, 117](#)
- EventCounter1End
 - [_quickSpin, 117](#)
- EventCounter1EndFrameID
 - [_quickSpin, 117](#)
- EventCounter1EndTimestamp
 - [_quickSpin, 117](#)
- EventCounter1Start
 - [_quickSpin, 117](#)
- EventCounter1StartFrameID
 - [_quickSpin, 118](#)
- EventCounter1StartTimestamp
 - [_quickSpin, 118](#)
- EventEncoder0Restarted
 - [_quickSpin, 118](#)
- EventEncoder0RestartedFrameID
 - [_quickSpin, 118](#)
- EventEncoder0RestartedTimestamp
 - [_quickSpin, 118](#)
- EventEncoder0Stopped
 - [_quickSpin, 118](#)
- EventEncoder0StoppedFrameID
 - [_quickSpin, 118](#)
- EventEncoder0StoppedTimestamp
 - [_quickSpin, 118](#)
- EventEncoder1Restarted
 - [_quickSpin, 119](#)
- EventEncoder1RestartedFrameID
 - [_quickSpin, 119](#)
- EventEncoder1RestartedTimestamp
 - [_quickSpin, 119](#)
- EventEncoder1Stopped
 - [_quickSpin, 119](#)
- EventEncoder1StoppedFrameID
 - [_quickSpin, 119](#)
- EventEncoder1StoppedTimestamp
 - [_quickSpin, 119](#)
- EventError
 - [_quickSpin, 119](#)
- EventErrorCode
 - [_quickSpin, 119](#)
- EventErrorFrameID
 - [_quickSpin, 120](#)
- EventErrorTimestamp
 - [_quickSpin, 120](#)
- EventExposureEnd
 - [_quickSpin, 120](#)
- EventExposureEndFrameID
 - [_quickSpin, 120](#)
- EventExposureEndTimestamp
 - [_quickSpin, 120](#)
- EventExposureStart
 - [_quickSpin, 120](#)
- EventExposureStartFrameID
 - [_quickSpin, 120](#)
- EventExposureStartTimestamp
 - [_quickSpin, 120](#)

[_quickSpin](#), 120
 EventFrameBurstEnd
 [_quickSpin](#), 121
 EventFrameBurstEndFrameID
 [_quickSpin](#), 121
 EventFrameBurstEndTimestamp
 [_quickSpin](#), 121
 EventFrameBurstStart
 [_quickSpin](#), 121
 EventFrameBurstStartFrameID
 [_quickSpin](#), 121
 EventFrameBurstStartTimestamp
 [_quickSpin](#), 121
 EventFrameEnd
 [_quickSpin](#), 121
 EventFrameEndFrameID
 [_quickSpin](#), 121
 EventFrameEndTimestamp
 [_quickSpin](#), 122
 EventFrameStart
 [_quickSpin](#), 122
 EventFrameStartFrameID
 [_quickSpin](#), 122
 EventFrameStartTimestamp
 [_quickSpin](#), 122
 EventFrameTransferEnd
 [_quickSpin](#), 122
 EventFrameTransferEndFrameID
 [_quickSpin](#), 122
 EventFrameTransferEndTimestamp
 [_quickSpin](#), 122
 EventFrameTransferStart
 [_quickSpin](#), 122
 EventFrameTransferStartFrameID
 [_quickSpin](#), 123
 EventFrameTransferStartTimestamp
 [_quickSpin](#), 123
 EventFrameTrigger
 [_quickSpin](#), 123
 EventFrameTriggerFrameID
 [_quickSpin](#), 123
 EventFrameTriggerTimestamp
 [_quickSpin](#), 123
 EventLine0AnyEdge
 [_quickSpin](#), 123
 EventLine0AnyEdgeFrameID
 [_quickSpin](#), 123
 EventLine0AnyEdgeTimestamp
 [_quickSpin](#), 123
 EventLine0FallingEdge
 [_quickSpin](#), 124
 EventLine0FallingEdgeFrameID
 [_quickSpin](#), 124
 EventLine0FallingEdgeTimestamp
 [_quickSpin](#), 124
 EventLine0RisingEdge
 [_quickSpin](#), 124
 EventLine0RisingEdgeFrameID
 [_quickSpin](#), 124
 EventLine0RisingEdgeTimestamp
 [_quickSpin](#), 124
 EventLine1AnyEdge
 [_quickSpin](#), 124
 EventLine1AnyEdgeFrameID
 [_quickSpin](#), 124
 EventLine1AnyEdgeTimestamp
 [_quickSpin](#), 125
 EventLine1FallingEdge
 [_quickSpin](#), 125
 EventLine1FallingEdgeFrameID
 [_quickSpin](#), 125
 EventLine1FallingEdgeTimestamp
 [_quickSpin](#), 125
 EventLine1RisingEdge
 [_quickSpin](#), 125
 EventLine1RisingEdgeFrameID
 [_quickSpin](#), 125
 EventLine1RisingEdgeTimestamp
 [_quickSpin](#), 125
 EventLinkSpeedChange
 [_quickSpin](#), 125
 EventLinkSpeedChangeFrameID
 [_quickSpin](#), 126
 EventLinkSpeedChangeTimestamp
 [_quickSpin](#), 126
 EventLinkTrigger0
 [_quickSpin](#), 126
 EventLinkTrigger0FrameID
 [_quickSpin](#), 126
 EventLinkTrigger0Timestamp
 [_quickSpin](#), 126
 EventLinkTrigger1
 [_quickSpin](#), 126
 EventLinkTrigger1FrameID
 [_quickSpin](#), 126
 EventLinkTrigger1Timestamp
 [_quickSpin](#), 126
 EventNotification
 [_quickSpin](#), 127
 EventNotification_Off
 CameraDefsC.h, 277
 EventNotification_On
 CameraDefsC.h, 277
 EventSelector
 [_quickSpin](#), 127
 EventSelector_Error
 CameraDefsC.h, 277
 EventSelector_ExposureEnd
 CameraDefsC.h, 277
 EventSelector_SerialPortReceive
 CameraDefsC.h, 277
 EventSequencerSetChange
 [_quickSpin](#), 127
 EventSequencerSetChangeFrameID
 [_quickSpin](#), 127
 EventSequencerSetChangeTimestamp

- [_quickSpin, 127](#)
- EventSerialData
 - [_quickSpin, 127](#)
- EventSerialDataLength
 - [_quickSpin, 127](#)
- EventSerialPortReceive
 - [_quickSpin, 127](#)
- EventSerialPortReceiveTimestamp
 - [_quickSpin, 128](#)
- EventSerialReceiveOverflow
 - [_quickSpin, 128](#)
- EventStream0TransferBlockEnd
 - [_quickSpin, 128](#)
- EventStream0TransferBlockEndFrameID
 - [_quickSpin, 128](#)
- EventStream0TransferBlockEndTimestamp
 - [_quickSpin, 128](#)
- EventStream0TransferBlockStart
 - [_quickSpin, 128](#)
- EventStream0TransferBlockStartFrameID
 - [_quickSpin, 128](#)
- EventStream0TransferBlockStartTimestamp
 - [_quickSpin, 128](#)
- EventStream0TransferBlockTrigger
 - [_quickSpin, 129](#)
- EventStream0TransferBlockTriggerFrameID
 - [_quickSpin, 129](#)
- EventStream0TransferBlockTriggerTimestamp
 - [_quickSpin, 129](#)
- EventStream0TransferBurstEnd
 - [_quickSpin, 129](#)
- EventStream0TransferBurstEndFrameID
 - [_quickSpin, 129](#)
- EventStream0TransferBurstEndTimestamp
 - [_quickSpin, 129](#)
- EventStream0TransferBurstStart
 - [_quickSpin, 129](#)
- EventStream0TransferBurstStartFrameID
 - [_quickSpin, 129](#)
- EventStream0TransferBurstStartTimestamp
 - [_quickSpin, 130](#)
- EventStream0TransferEnd
 - [_quickSpin, 130](#)
- EventStream0TransferEndFrameID
 - [_quickSpin, 130](#)
- EventStream0TransferEndTimestamp
 - [_quickSpin, 130](#)
- EventStream0TransferOverflow
 - [_quickSpin, 130](#)
- EventStream0TransferOverflowFrameID
 - [_quickSpin, 130](#)
- EventStream0TransferOverflowTimestamp
 - [_quickSpin, 130](#)
- EventStream0TransferPause
 - [_quickSpin, 130](#)
- EventStream0TransferPauseFrameID
 - [_quickSpin, 131](#)
- EventStream0TransferPauseTimestamp
 - [_quickSpin, 131](#)
- EventStream0TransferResume
 - [_quickSpin, 131](#)
- EventStream0TransferResumeFrameID
 - [_quickSpin, 131](#)
- EventStream0TransferResumeTimestamp
 - [_quickSpin, 131](#)
- EventStream0TransferStart
 - [_quickSpin, 131](#)
- EventStream0TransferStartFrameID
 - [_quickSpin, 131](#)
- EventStream0TransferStartTimestamp
 - [_quickSpin, 131](#)
- EventTest
 - [_quickSpin, 132](#)
- EventTestTimestamp
 - [_quickSpin, 132](#)
- EventTimer0End
 - [_quickSpin, 132](#)
- EventTimer0EndFrameID
 - [_quickSpin, 132](#)
- EventTimer0EndTimestamp
 - [_quickSpin, 132](#)
- EventTimer0Start
 - [_quickSpin, 132](#)
- EventTimer0StartFrameID
 - [_quickSpin, 132](#)
- EventTimer0StartTimestamp
 - [_quickSpin, 132](#)
- EventTimer1End
 - [_quickSpin, 133](#)
- EventTimer1EndFrameID
 - [_quickSpin, 133](#)
- EventTimer1EndTimestamp
 - [_quickSpin, 133](#)
- EventTimer1Start
 - [_quickSpin, 133](#)
- EventTimer1StartFrameID
 - [_quickSpin, 133](#)
- EventTimer1StartTimestamp
 - [_quickSpin, 133](#)
- Expert
 - SpinnakerGenApiDefsC.h, [502](#)
- ExposureActiveMode
 - [_quickSpin, 133](#)
- ExposureActiveMode_AllPixels
 - CameraDefsC.h, [277](#)
- ExposureActiveMode_AnyPixels
 - CameraDefsC.h, [277](#)
- ExposureActiveMode_Line1
 - CameraDefsC.h, [277](#)
- ExposureAuto
 - [_quickSpin, 133](#)
- ExposureAuto_Continuous
 - CameraDefsC.h, [278](#)
- ExposureAuto_Off
 - CameraDefsC.h, [278](#)
- ExposureAuto_Once

- CameraDefsC.h, [278](#)
- ExposureMode
 - _quickSpin, [134](#)
- ExposureMode_Timed
 - CameraDefsC.h, [278](#)
- ExposureMode_TriggerWidth
 - CameraDefsC.h, [278](#)
- ExposureTime
 - _quickSpin, [134](#)
- ExposureTimeMode
 - _quickSpin, [134](#)
- ExposureTimeMode_Common
 - CameraDefsC.h, [278](#)
- ExposureTimeMode_Individual
 - CameraDefsC.h, [278](#)
- ExposureTimeSelector
 - _quickSpin, [134](#)
- ExposureTimeSelector_Blue
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Common
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Cyan
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Green
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Infrared
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Magenta
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Red
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Stage1
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Stage2
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Ultraviolet
 - CameraDefsC.h, [279](#)
- ExposureTimeSelector_Yellow
 - CameraDefsC.h, [279](#)
- FactoryReset
 - _quickSpin, [134](#)
- False
 - SpinnakerDefsC.h, [451](#)
- FileAccessBuffer
 - _quickSpin, [134](#)
- FileAccessLength
 - _quickSpin, [134](#)
- FileAccessOffset
 - _quickSpin, [134](#)
- FileOpenMode
 - _quickSpin, [135](#)
- FileOpenMode_Read
 - CameraDefsC.h, [279](#)
- FileOpenMode_ReadWrite
 - CameraDefsC.h, [279](#)
- FileOpenMode_Write
 - CameraDefsC.h, [279](#)
- FileOperationExecute
 - _quickSpin, [135](#)
- FileOperationResult
 - _quickSpin, [135](#)
- FileOperationSelector
 - _quickSpin, [135](#)
- FileOperationSelector_Close
 - CameraDefsC.h, [280](#)
- FileOperationSelector_Delete
 - CameraDefsC.h, [280](#)
- FileOperationSelector_Open
 - CameraDefsC.h, [280](#)
- FileOperationSelector_Read
 - CameraDefsC.h, [280](#)
- FileOperationSelector_Write
 - CameraDefsC.h, [280](#)
- FileOperationStatus
 - _quickSpin, [135](#)
- FileOperationStatus_Failure
 - CameraDefsC.h, [280](#)
- FileOperationStatus_Overflow
 - CameraDefsC.h, [280](#)
- FileOperationStatus_Success
 - CameraDefsC.h, [280](#)
- FileSelector
 - _quickSpin, [135](#)
- FileSelector_SerialPort0
 - CameraDefsC.h, [280](#)
- FileSelector_UserFile1
 - CameraDefsC.h, [280](#)
- FileSelector_UserSet0
 - CameraDefsC.h, [280](#)
- FileSelector_UserSet1
 - CameraDefsC.h, [280](#)
- FileSelector_UserSetDefault
 - CameraDefsC.h, [280](#)
- FileSize
 - _quickSpin, [135](#)
- FilterDriverStatus
 - _quickSpinTLInterface, [175](#)
- FilterDriverStatus_Disabled
 - TransportLayerDefsC.h, [511](#)
- FilterDriverStatus_Enabled
 - TransportLayerDefsC.h, [511](#)
- FilterDriverStatus_NotSupported
 - TransportLayerDefsC.h, [511](#)
- fixedIncrement
 - SpinnakerGenApiDefsC.h, [498](#)
- FloatNode
 - SpinnakerGenApiDefsC.h, [500](#)
- fnAutomatic
 - SpinnakerGenApiDefsC.h, [497](#)
- fnFixed
 - SpinnakerGenApiDefsC.h, [497](#)
- fnScientific
 - SpinnakerGenApiDefsC.h, [497](#)
- frameRate
 - _spinAVIOption, [190](#)
 - _spinH264Option, [198](#)

- [_spinMJPGOption, 203](#)
- Gain
 - [_quickSpin, 135](#)
- GainAuto
 - [_quickSpin, 136](#)
- GainAuto_Continuous
 - [CameraDefsC.h, 282](#)
- GainAuto_Off
 - [CameraDefsC.h, 282](#)
- GainAuto_Once
 - [CameraDefsC.h, 282](#)
- GainAutoBalance
 - [_quickSpin, 136](#)
- GainAutoBalance_Continuous
 - [CameraDefsC.h, 282](#)
- GainAutoBalance_Off
 - [CameraDefsC.h, 282](#)
- GainAutoBalance_Once
 - [CameraDefsC.h, 282](#)
- GainSelector
 - [_quickSpin, 136](#)
- GainSelector_All
 - [CameraDefsC.h, 282](#)
- Gamma
 - [_quickSpin, 136](#)
- GammaEnable
 - [_quickSpin, 136](#)
- GenlCamXMLLocation
 - [_quickSpinTLDevice, 170](#)
- GenlCamXMLLocation_Device
 - [TransportLayerDefsC.h, 512](#)
- GenlCamXMLLocation_Host
 - [TransportLayerDefsC.h, 512](#)
- GenlCamXMLPath
 - [_quickSpinTLDevice, 170](#)
- GenTLFNCVersionMajor
 - [_quickSpinTLSystem, 187](#)
- GenTLFNCVersionMinor
 - [_quickSpinTLSystem, 187](#)
- GenTLFNCVersionSubMinor
 - [_quickSpinTLSystem, 187](#)
- GenTLVersionMajor
 - [_quickSpinTLSystem, 187](#)
- GenTLVersionMinor
 - [_quickSpinTLSystem, 187](#)
- GEV
 - [SpinnakerGenApiDefsC.h, 502](#)
- GevActionDeviceKey
 - [_quickSpinTLInterface, 176](#)
- GevActionGroupKey
 - [_quickSpinTLInterface, 176](#)
- GevActionGroupMask
 - [_quickSpinTLInterface, 176](#)
- GevActionTime
 - [_quickSpinTLInterface, 176](#)
- GevActiveLinkCount
 - [_quickSpin, 136](#)
- GevCCP
 - [_quickSpin, 136](#)
 - [_quickSpinTLDevice, 170](#)
- GevCCP_ControlAccess
 - [CameraDefsC.h, 283](#)
- GevCCP_EnumEntry_GevCCP_ControlAccess
 - [TransportLayerDefsC.h, 512](#)
- GevCCP_EnumEntry_GevCCP_ExclusiveAccess
 - [TransportLayerDefsC.h, 512](#)
- GevCCP_EnumEntry_GevCCP_OpenAccess
 - [TransportLayerDefsC.h, 512](#)
- GevCCP_ExclusiveAccess
 - [CameraDefsC.h, 283](#)
- GevCCP_OpenAccess
 - [CameraDefsC.h, 283](#)
- GevCurrentDefaultGateway
 - [_quickSpin, 136](#)
- GevCurrentIPAddress
 - [_quickSpin, 137](#)
- GevCurrentIPConfigurationDHCP
 - [_quickSpin, 137](#)
- GevCurrentIPConfigurationLLA
 - [_quickSpin, 137](#)
- GevCurrentIPConfigurationPersistentIP
 - [_quickSpin, 137](#)
- GevCurrentPhysicalLinkConfiguration
 - [_quickSpin, 137](#)
- GevCurrentPhysicalLinkConfiguration_DynamicLAG
 - [CameraDefsC.h, 283](#)
- GevCurrentPhysicalLinkConfiguration_MultiLink
 - [CameraDefsC.h, 283](#)
- GevCurrentPhysicalLinkConfiguration_SingleLink
 - [CameraDefsC.h, 283](#)
- GevCurrentPhysicalLinkConfiguration_StaticLAG
 - [CameraDefsC.h, 283](#)
- GevCurrentSubnetMask
 - [_quickSpin, 137](#)
- GevDeviceAutoForceIP
 - [_quickSpinTLDevice, 170](#)
 - [_quickSpinTLInterface, 176](#)
- GevDeviceDiscoverMaximumPacketSize
 - [_quickSpinTLDevice, 171](#)
- GevDeviceForceGateway
 - [_quickSpinTLDevice, 171](#)
 - [_quickSpinTLInterface, 176](#)
- GevDeviceForceIP
 - [_quickSpinTLDevice, 171](#)
 - [_quickSpinTLInterface, 176](#)
- GevDeviceForceIPAddress
 - [_quickSpinTLDevice, 171](#)
 - [_quickSpinTLInterface, 176](#)
- GevDeviceForceSubnetMask
 - [_quickSpinTLDevice, 171](#)
 - [_quickSpinTLInterface, 177](#)
- GevDeviceGateway
 - [_quickSpinTLDevice, 171](#)
 - [_quickSpinTLInterface, 177](#)
- GevDeviceIPAddress
 - [_quickSpinTLDevice, 171](#)

- [_quickSpinTLInterface, 177](#)
- GevDeviceIsWrongSubnet
 - [_quickSpinTLDevice, 171](#)
- GevDeviceMACAddress
 - [_quickSpinTLDevice, 172](#)
 - [_quickSpinTLInterface, 177](#)
- GevDeviceMaximumPacketSize
 - [_quickSpinTLDevice, 172](#)
- GevDeviceMaximumRetryCount
 - [_quickSpinTLDevice, 172](#)
- GevDeviceModelsBigEndian
 - [_quickSpinTLDevice, 172](#)
- GevDevicePort
 - [_quickSpinTLDevice, 172](#)
- GevDeviceReadAndWriteTimeout
 - [_quickSpinTLDevice, 172](#)
- GevDeviceSubnetMask
 - [_quickSpinTLDevice, 172](#)
 - [_quickSpinTLInterface, 177](#)
- GevDiscoveryAckDelay
 - [_quickSpin, 137](#)
- GevFirstURL
 - [_quickSpin, 137](#)
- GevGVCPExtendedStatusCodes
 - [_quickSpin, 138](#)
- GevGVCPExtendedStatusCodesSelector
 - [_quickSpin, 138](#)
- GevGVCPExtendedStatusCodesSelector_Version1_1
 - [CameraDefsC.h, 283](#)
- GevGVCPExtendedStatusCodesSelector_Version2_0
 - [CameraDefsC.h, 283](#)
- GevGVCPHeartbeatDisable
 - [_quickSpin, 138](#)
- GevGVCPPendingAck
 - [_quickSpin, 138](#)
- GevGVCPPendingTimeout
 - [_quickSpin, 138](#)
- GevGVSPExtendedIDMode
 - [_quickSpin, 138](#)
- GevGVSPExtendedIDMode_Off
 - [CameraDefsC.h, 284](#)
- GevGVSPExtendedIDMode_On
 - [CameraDefsC.h, 284](#)
- GevHeartbeatTimeout
 - [_quickSpin, 138](#)
- GevIEEE1588
 - [_quickSpin, 138](#)
- GevIEEE1588ClockAccuracy
 - [_quickSpin, 139](#)
- GevIEEE1588ClockAccuracy_Unknown
 - [CameraDefsC.h, 284](#)
- GevIEEE1588Mode
 - [_quickSpin, 139](#)
- GevIEEE1588Mode_Auto
 - [CameraDefsC.h, 284](#)
- GevIEEE1588Mode_SlaveOnly
 - [CameraDefsC.h, 284](#)
- GevIEEE1588Status
 - [_quickSpin, 139](#)
- GevIEEE1588Status_Disabled
 - [CameraDefsC.h, 284](#)
- GevIEEE1588Status_Faulty
 - [CameraDefsC.h, 284](#)
- GevIEEE1588Status_Initializing
 - [CameraDefsC.h, 284](#)
- GevIEEE1588Status_Listening
 - [CameraDefsC.h, 285](#)
- GevIEEE1588Status_Master
 - [CameraDefsC.h, 285](#)
- GevIEEE1588Status_Passive
 - [CameraDefsC.h, 285](#)
- GevIEEE1588Status_PreMaster
 - [CameraDefsC.h, 285](#)
- GevIEEE1588Status_Slave
 - [CameraDefsC.h, 285](#)
- GevIEEE1588Status_Uncalibrated
 - [CameraDefsC.h, 285](#)
- GevInterfaceDefaultGateway
 - [_quickSpinTLSystem, 187](#)
- GevInterfaceDefaultIPAddress
 - [_quickSpinTLSystem, 187](#)
- GevInterfaceDefaultSubnetMask
 - [_quickSpinTLSystem, 187](#)
- GevInterfaceGateway
 - [_quickSpinTLInterface, 177](#)
- GevInterfaceGatewaySelector
 - [_quickSpinTLInterface, 177](#)
- GevInterfaceMACAddress
 - [_quickSpinTLInterface, 177](#)
 - [_quickSpinTLSystem, 188](#)
- GevInterfaceMTU
 - [_quickSpinTLInterface, 178](#)
- GevInterfaceReceiveLinkSpeed
 - [_quickSpinTLInterface, 178](#)
- GevInterfaceSelector
 - [_quickSpin, 139](#)
- GevInterfaceSubnetIPAddress
 - [_quickSpinTLInterface, 178](#)
- GevInterfaceSubnetMask
 - [_quickSpinTLInterface, 178](#)
- GevInterfaceSubnetSelector
 - [_quickSpinTLInterface, 178](#)
- GevInterfaceTransmitLinkSpeed
 - [_quickSpinTLInterface, 178](#)
- GevIPConfigurationStatus
 - [_quickSpin, 139](#)
- GevIPConfigurationStatus_DHCP
 - [CameraDefsC.h, 285](#)
- GevIPConfigurationStatus_ForceIP
 - [CameraDefsC.h, 285](#)
- GevIPConfigurationStatus_LLA
 - [CameraDefsC.h, 285](#)
- GevIPConfigurationStatus_None
 - [CameraDefsC.h, 285](#)
- GevIPConfigurationStatus_PersistentIP
 - [CameraDefsC.h, 285](#)

- GevMACAddress
 - [_quickSpin, 139](#)
- GevMCDA
 - [_quickSpin, 139](#)
- GevMCPHostPort
 - [_quickSpin, 139](#)
- GevMCRC
 - [_quickSpin, 140](#)
- GevMCSP
 - [_quickSpin, 140](#)
- GevMCTT
 - [_quickSpin, 140](#)
- GevNumberOfInterfaces
 - [_quickSpin, 140](#)
- GevPAUSEFrameReception
 - [_quickSpin, 140](#)
- GevPAUSEFrameTransmission
 - [_quickSpin, 140](#)
- GevPersistentDefaultGateway
 - [_quickSpin, 140](#)
- GevPersistentIPAddress
 - [_quickSpin, 140](#)
- GevPersistentSubnetMask
 - [_quickSpin, 141](#)
- GevPhysicalLinkConfiguration
 - [_quickSpin, 141](#)
- GevPhysicalLinkConfiguration_DynamicLAG
 - [CameraDefsC.h, 285](#)
- GevPhysicalLinkConfiguration_MultiLink
 - [CameraDefsC.h, 285](#)
- GevPhysicalLinkConfiguration_SingleLink
 - [CameraDefsC.h, 285](#)
- GevPhysicalLinkConfiguration_StaticLAG
 - [CameraDefsC.h, 285](#)
- GevPrimaryApplicationIPAddress
 - [_quickSpin, 141](#)
- GevPrimaryApplicationSocket
 - [_quickSpin, 141](#)
- GevPrimaryApplicationSwitchoverKey
 - [_quickSpin, 141](#)
- GevSCCFGAllInTransmission
 - [_quickSpin, 141](#)
- GevSCCFGExtendedChunkData
 - [_quickSpin, 141](#)
- GevSCCFGPacketResendDestination
 - [_quickSpin, 141](#)
- GevSCCFGUnconditionalStreaming
 - [_quickSpin, 142](#)
- GevSCDA
 - [_quickSpin, 142](#)
- GevSCPD
 - [_quickSpin, 142](#)
- GevSCPDDirection
 - [_quickSpin, 142](#)
- GevSCPHostPort
 - [_quickSpin, 142](#)
- GevSCPIInterfaceIndex
 - [_quickSpin, 142](#)
- GevSCPSBigEndian
 - [_quickSpin, 142](#)
- GevSCPSDoNotFragment
 - [_quickSpin, 142](#)
- GevSCPSFireTestPacket
 - [_quickSpin, 143](#)
- GevSCPSPacketSize
 - [_quickSpin, 143](#)
- GevSCSP
 - [_quickSpin, 143](#)
- GevSCZoneConfigurationLock
 - [_quickSpin, 143](#)
- GevSCZoneCount
 - [_quickSpin, 143](#)
- GevSCZoneDirectionAll
 - [_quickSpin, 143](#)
- GevSecondURL
 - [_quickSpin, 143](#)
- GevStreamChannelSelector
 - [_quickSpin, 143](#)
- GevSupportedOption
 - [_quickSpin, 144](#)
- GevSupportedOptionSelector
 - [_quickSpin, 144](#)
- GevSupportedOptionSelector_Action
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_CCPApplicationSocket
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_CommandsConcatenation
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_DiscoveryAckDelay
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_DiscoveryAckDelayWritable
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_Event
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_EventData
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_ExtendedStatusCodes
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_HeartbeatDisable
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_IPConfigurationDHCP
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_IPConfigurationLLA
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_IPConfigurationPersistentIP
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_LinkSpeed
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_ManifestTable
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_MessageChannelSourceSocket
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_PacketResend
 - [CameraDefsC.h, 286](#)
- GevSupportedOptionSelector_PendingAck
 - [CameraDefsC.h, 286](#)

- GevSupportedOptionSelector_SerialNumber
 - CameraDefsC.h, [286](#)
- GevSupportedOptionSelector_StreamChannelSourceSocket
 - CameraDefsC.h, [286](#)
- GevSupportedOptionSelector_TestData
 - CameraDefsC.h, [286](#)
- GevSupportedOptionSelector_UserDefinedName
 - CameraDefsC.h, [286](#)
- GevSupportedOptionSelector_WriteMem
 - CameraDefsC.h, [286](#)
- GevTimestampTickFrequency
 - _quickSpin, [144](#)
- GevVersionMajor
 - _quickSpinTLDevice, [172](#)
 - _quickSpinTLSystem, [188](#)
- GevVersionMinor
 - _quickSpinTLDevice, [173](#)
 - _quickSpinTLSystem, [188](#)
- GUIXMLLocation
 - _quickSpinTLDevice, [173](#)
- GUIXMLLocation_Device
 - TransportLayerDefsC.h, [512](#)
- GUIXMLLocation_Host
 - TransportLayerDefsC.h, [512](#)
- GuiXmlManifestAddress
 - _quickSpin, [144](#)
- GUIXMLPath
 - _quickSpinTLDevice, [173](#)
- Guru
 - SpinnakerGenApiDefsC.h, [502](#)
- Height
 - _quickSpin, [144](#)
- height
 - _spinAVIOption, [190](#)
 - _spinH264Option, [199](#)
 - _spinMJPGOption, [203](#)
- HeightMax
 - _quickSpin, [144](#)
- HexNumber
 - SpinnakerGenApiDefsC.h, [501](#)
- HostAdapterDriverVersion
 - _quickSpinTLInterface, [178](#)
- HostAdapterName
 - _quickSpinTLInterface, [178](#)
- HostAdapterVendor
 - _quickSpinTLInterface, [179](#)
- IBoolean Access, [56](#)
- ICategory Access, [58](#)
- ICommand Access, [57](#)
- idFrom
 - SpinnakerGenApiDefsC.h, [498](#)
- idNone
 - SpinnakerGenApiDefsC.h, [498](#)
- idTo
 - SpinnakerGenApiDefsC.h, [498](#)
- IEnumEntry Access, [55](#)
- IEnumeration Access, [54](#)
- IFloat Access, [53](#)
- IIDC
 - SpinnakerGenApiDefsC.h, [502](#)
- IInteger Access, [52](#)
- Image Access, [34](#)
- Image Processor Access, [35](#)
- ImageComponentEnable
 - _quickSpin, [144](#)
- ImageComponentSelector
 - _quickSpin, [144](#)
- ImageComponentSelector_Color
 - CameraDefsC.h, [286](#)
- ImageComponentSelector_Confidence
 - CameraDefsC.h, [287](#)
- ImageComponentSelector_Disparity
 - CameraDefsC.h, [287](#)
- ImageComponentSelector_Infrared
 - CameraDefsC.h, [286](#)
- ImageComponentSelector_Intensity
 - CameraDefsC.h, [286](#)
- ImageComponentSelector_Range
 - CameraDefsC.h, [286](#)
- ImageComponentSelector_Scatter
 - CameraDefsC.h, [287](#)
- ImageComponentSelector_Ultraviolet
 - CameraDefsC.h, [286](#)
- ImageCompressionBitrate
 - _quickSpin, [145](#)
- ImageCompressionJPEGFormatOption
 - _quickSpin, [145](#)
- ImageCompressionJPEGFormatOption_BaselineOptimized
 - CameraDefsC.h, [287](#)
- ImageCompressionJPEGFormatOption_BaselineStandard
 - CameraDefsC.h, [287](#)
- ImageCompressionJPEGFormatOption_Lossless
 - CameraDefsC.h, [287](#)
- ImageCompressionJPEGFormatOption_Progressive
 - CameraDefsC.h, [287](#)
- ImageCompressionMode
 - _quickSpin, [145](#)
- ImageCompressionMode_Lossless
 - CameraDefsC.h, [288](#)
- ImageCompressionMode_Off
 - CameraDefsC.h, [288](#)
- ImageCompressionQuality
 - _quickSpin, [145](#)
- ImageCompressionRateOption
 - _quickSpin, [145](#)
- ImageCompressionRateOption_FixBitrate
 - CameraDefsC.h, [288](#)
- ImageCompressionRateOption_FixQuality
 - CameraDefsC.h, [288](#)
- ImageList Access, [31](#)
- ImageStatistics Access, [39](#)
- include/spinc/CameraDefsC.h, [209](#)
- include/spinc/ChunkDataDefC.h, [326](#)
- include/spinc/QuickSpinC.h, [327](#)
- include/spinc/QuickSpinDefsC.h, [329](#)

- include/spinc/SpinnakerC.h, [331](#)
- include/spinc/SpinnakerDefsC.h, [436](#)
- include/spinc/SpinnakerGenApiC.h, [452](#)
- include/spinc/SpinnakerGenApiDefsC.h, [493](#)
- include/spinc/SpinnakerPlatformC.h, [504](#)
- include/spinc/SpinVideoC.h, [505](#)
- include/spinc/TransportLayerDefsC.h, [507](#)
- include/spinc/TransportLayerDeviceC.h, [515](#)
- include/spinc/TransportLayerInterfaceC.h, [516](#)
- include/spinc/TransportLayerStreamC.h, [517](#)
- include/spinc/TransportLayerSystemC.h, [517](#)
- IncompatibleDeviceCount
 - _quickSpinTLInterface, [179](#)
- IncompatibleDeviceID
 - _quickSpinTLInterface, [179](#)
- IncompatibleDeviceModelName
 - _quickSpinTLInterface, [179](#)
- IncompatibleDeviceSelector
 - _quickSpinTLInterface, [179](#)
- IncompatibleDeviceVendorName
 - _quickSpinTLInterface, [179](#)
- IncompatibleGevDeviceIPAddress
 - _quickSpinTLInterface, [179](#)
- IncompatibleGevDeviceMACAddress
 - _quickSpinTLInterface, [179](#)
- IncompatibleGevDeviceSubnetMask
 - _quickSpinTLInterface, [180](#)
- Increasing
 - SpinnakerGenApiDefsC.h, [502](#)
- indexedColor_8bit
 - _spinBMPOption, [191](#)
- IntegerNode
 - SpinnakerGenApiDefsC.h, [500](#)
- Interface Access, [32](#)
- InterfaceDisplayName
 - _quickSpinTLInterface, [180](#)
 - _quickSpinTLSystem, [188](#)
- InterfaceID
 - _quickSpinTLInterface, [180](#)
 - _quickSpinTLSystem, [188](#)
- InterfaceList Access, [29](#)
- InterfaceSelector
 - _quickSpinTLSystem, [188](#)
- InterfaceType
 - _quickSpinTLInterface, [180](#)
- InterfaceType_CameraLink
 - TransportLayerDefsC.h, [512](#)
- InterfaceType_CameraLinkHS
 - TransportLayerDefsC.h, [513](#)
- InterfaceType_CoaXPRESS
 - TransportLayerDefsC.h, [513](#)
- InterfaceType_Custom
 - TransportLayerDefsC.h, [513](#)
- InterfaceType_GigEVision
 - TransportLayerDefsC.h, [512](#)
- InterfaceType_USB3Vision
 - TransportLayerDefsC.h, [513](#)
- InterfaceUpdateList
 - _quickSpinTLSystem, [188](#)
- interlaced
 - _spinPNGOption, [205](#)
- intfIBase
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIBoolean
 - SpinnakerGenApiDefsC.h, [499](#)
- intfICategory
 - SpinnakerGenApiDefsC.h, [499](#)
- intfICommand
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIEnumEntry
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIEnumeration
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIFloat
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIInteger
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIPort
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIRegister
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIString
 - SpinnakerGenApiDefsC.h, [499](#)
- intfIValue
 - SpinnakerGenApiDefsC.h, [499](#)
- Invisible
 - SpinnakerGenApiDefsC.h, [502](#)
- IPv4Address
 - SpinnakerGenApiDefsC.h, [501](#)
- IRegister Access, [59](#)
- IspEnable
 - _quickSpin, [145](#)
- IValue Access, [50](#)
- Linear
 - SpinnakerGenApiDefsC.h, [501](#)
- LineFilterWidth
 - _quickSpin, [145](#)
- LineFormat
 - _quickSpin, [145](#)
- LineFormat_LVDS
 - CameraDefsC.h, [288](#)
- LineFormat_NoConnect
 - CameraDefsC.h, [288](#)
- LineFormat_OpenDrain
 - CameraDefsC.h, [288](#)
- LineFormat_OptoCoupled
 - CameraDefsC.h, [288](#)
- LineFormat_RS422
 - CameraDefsC.h, [288](#)
- LineFormat_TriState
 - CameraDefsC.h, [288](#)
- LineFormat_TTL
 - CameraDefsC.h, [288](#)
- LineInputFilterSelector
 - _quickSpin, [146](#)
- LineInputFilterSelector_Debounce

- CameraDefsC.h, [289](#)
- LineInputFilterSelector_Deglitch
 - CameraDefsC.h, [289](#)
- LineInverter
 - _quickSpin, [146](#)
- LineMode
 - _quickSpin, [146](#)
- LineMode_Input
 - CameraDefsC.h, [289](#)
- LineMode_Output
 - CameraDefsC.h, [289](#)
- LinePitch
 - _quickSpin, [146](#)
- LineSelector
 - _quickSpin, [146](#)
- LineSelector_Line0
 - CameraDefsC.h, [289](#)
- LineSelector_Line1
 - CameraDefsC.h, [289](#)
- LineSelector_Line2
 - CameraDefsC.h, [289](#)
- LineSelector_Line3
 - CameraDefsC.h, [289](#)
- LineSource
 - _quickSpin, [146](#)
- LineSource_AllPixel
 - CameraDefsC.h, [290](#)
- LineSource_AnyPixel
 - CameraDefsC.h, [290](#)
- LineSource_Counter0Active
 - CameraDefsC.h, [290](#)
- LineSource_Counter1Active
 - CameraDefsC.h, [290](#)
- LineSource_ExposureActive
 - CameraDefsC.h, [290](#)
- LineSource_FrameTriggerWait
 - CameraDefsC.h, [290](#)
- LineSource_Line0
 - CameraDefsC.h, [290](#)
- LineSource_Line1
 - CameraDefsC.h, [290](#)
- LineSource_Line2
 - CameraDefsC.h, [290](#)
- LineSource_Line3
 - CameraDefsC.h, [290](#)
- LineSource_LogicBlock0
 - CameraDefsC.h, [290](#)
- LineSource_LogicBlock1
 - CameraDefsC.h, [290](#)
- LineSource_Off
 - CameraDefsC.h, [290](#)
- LineSource_PPSSignal
 - CameraDefsC.h, [290](#)
- LineSource_SerialPort0
 - CameraDefsC.h, [290](#)
- LineSource_UserOutput0
 - CameraDefsC.h, [290](#)
- LineSource_UserOutput1
 - CameraDefsC.h, [290](#)
- LineSource_UserOutput2
 - CameraDefsC.h, [290](#)
- LineSource_UserOutput3
 - CameraDefsC.h, [290](#)
- LineStatus
 - _quickSpin, [146](#)
- LineStatusAll
 - _quickSpin, [146](#)
- LinkErrorCount
 - _quickSpin, [147](#)
- LinkUptime
 - _quickSpin, [147](#)
- listIncrement
 - SpinnakerGenApiDefsC.h, [498](#)
- LittleEndian
 - SpinnakerGenApiDefsC.h, [498](#)
- Logarithmic
 - SpinnakerGenApiDefsC.h, [501](#)
- Logging Event Data Access, [40](#)
- LogicBlockLUTInputActivation
 - _quickSpin, [147](#)
- LogicBlockLUTInputActivation_AnyEdge
 - CameraDefsC.h, [290](#)
- LogicBlockLUTInputActivation_FallingEdge
 - CameraDefsC.h, [290](#)
- LogicBlockLUTInputActivation_LevelHigh
 - CameraDefsC.h, [290](#)
- LogicBlockLUTInputActivation_LevelLow
 - CameraDefsC.h, [290](#)
- LogicBlockLUTInputActivation_RisingEdge
 - CameraDefsC.h, [290](#)
- LogicBlockLUTInputSelector
 - _quickSpin, [147](#)
- LogicBlockLUTInputSelector_Input0
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSelector_Input1
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSelector_Input2
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSelector_Input3
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSource
 - _quickSpin, [147](#)
- LogicBlockLUTInputSource_AcquisitionActive
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSource_Counter0End
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSource_Counter0Start
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSource_Counter1End
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSource_Counter1Start
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSource_ExposureEnd
 - CameraDefsC.h, [291](#)
- LogicBlockLUTInputSource_ExposureStart
 - CameraDefsC.h, [291](#)

LogicBlockLUTInputSource_FrameTriggerWait
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_Line0
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_Line1
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_Line2
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_Line3
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_LogicBlock0
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_LogicBlock1
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_UserOutput0
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_UserOutput1
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_UserOutput2
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_UserOutput3
 CameraDefsC.h, [291](#)
 LogicBlockLUTInputSource_Zero
 CameraDefsC.h, [291](#)
 LogicBlockLUTOutputValue
 _quickSpin, [147](#)
 LogicBlockLUTOutputValueAll
 _quickSpin, [147](#)
 LogicBlockLUTRowIndex
 _quickSpin, [147](#)
 LogicBlockLUTSelector
 _quickSpin, [148](#)
 LogicBlockLUTSelector_Enable
 CameraDefsC.h, [292](#)
 LogicBlockLUTSelector_Value
 CameraDefsC.h, [292](#)
 LogicBlockSelector
 _quickSpin, [148](#)
 LogicBlockSelector_LogicBlock0
 CameraDefsC.h, [292](#)
 LogicBlockSelector_LogicBlock1
 CameraDefsC.h, [292](#)
 LUTEnable
 _quickSpin, [148](#)
 LUTIndex
 _quickSpin, [148](#)
 LUTSelector
 _quickSpin, [148](#)
 LUTSelector_LUT1
 CameraDefsC.h, [292](#)
 LUTValue
 _quickSpin, [148](#)
 LUTValueAll
 _quickSpin, [148](#)

 m_blackLevel
 _spinChunkData, [193](#)
 m_compressionMode
 _spinChunkData, [193](#)

 m_compressionRatio
 _spinChunkData, [193](#)
 m_counterValue
 _spinChunkData, [193](#)
 m_cRC
 _spinChunkData, [193](#)
 m_encoderValue
 _spinChunkData, [193](#)
 m_exposureEndLineStatusAll
 _spinChunkData, [193](#)
 m_exposureTime
 _spinChunkData, [194](#)
 m_frameID
 _spinChunkData, [194](#)
 m_gain
 _spinChunkData, [194](#)
 m_height
 _spinChunkData, [194](#)
 m_image
 _spinChunkData, [194](#)
 m_inferenceConfidence
 _spinChunkData, [194](#)
 m_inferenceFrameId
 _spinChunkData, [194](#)
 m_inferenceResult
 _spinChunkData, [194](#)
 m_linePitch
 _spinChunkData, [195](#)
 m_lineStatusAll
 _spinChunkData, [195](#)
 m_offsetX
 _spinChunkData, [195](#)
 m_offsetY
 _spinChunkData, [195](#)
 m_partSelector
 _spinChunkData, [195](#)
 m_pixelDynamicRangeMax
 _spinChunkData, [195](#)
 m_pixelDynamicRangeMin
 _spinChunkData, [195](#)
 m_scan3dAxisMax
 _spinChunkData, [195](#)
 m_scan3dAxisMin
 _spinChunkData, [196](#)
 m_scan3dCoordinateOffset
 _spinChunkData, [196](#)
 m_scan3dCoordinateReferenceValue
 _spinChunkData, [196](#)
 m_scan3dCoordinateScale
 _spinChunkData, [196](#)
 m_scan3dInvalidDataValue
 _spinChunkData, [196](#)
 m_scan3dTransformValue
 _spinChunkData, [196](#)
 m_scanLineSelector
 _spinChunkData, [196](#)
 m_sequencerSetActive
 _spinChunkData, [196](#)

- m_serialDataLength
 - _spinChunkData, [197](#)
- m_streamChannelID
 - _spinChunkData, [197](#)
- m_timerValue
 - _spinChunkData, [197](#)
- m_timestamp
 - _spinChunkData, [197](#)
- m_timestampLatchValue
 - _spinChunkData, [197](#)
- m_transferBlockID
 - _spinChunkData, [197](#)
- m_transferQueueCurrentBlockCount
 - _spinChunkData, [197](#)
- m_width
 - _spinChunkData, [197](#)
- MACAddress
 - SpinnakerGenApiDefsC.h, [501](#)
- major
 - _spinLibraryVersion, [202](#)
- MaxDeviceResetTime
 - _quickSpin, [148](#)
- minor
 - _spinLibraryVersion, [202](#)
- NA
 - SpinnakerGenApiDefsC.h, [497](#)
- NI
 - SpinnakerGenApiDefsC.h, [497](#)
- No
 - SpinnakerGenApiDefsC.h, [503](#)
- NoCache
 - SpinnakerGenApiDefsC.h, [497](#)
- Node Access, [49](#)
- Node Map Access, [48](#)
- noIncrement
 - SpinnakerGenApiDefsC.h, [498](#)
- None
 - SpinnakerGenApiDefsC.h, [502](#)
- NUM_ACQUISITIONMODE
 - CameraDefsC.h, [242](#)
- NUM_ACQUISITIONSTATUSSELECTOR
 - CameraDefsC.h, [242](#)
- NUM_ACTIONUNCONDITIONALMODE
 - CameraDefsC.h, [243](#)
- NUM_ADCBITDEPTH
 - CameraDefsC.h, [243](#)
- NUM_AUTOALGORITHMSELECTOR
 - CameraDefsC.h, [243](#)
- NUM_AUTOEXPOSURECONTROLPRIORITY
 - CameraDefsC.h, [244](#)
- NUM_AUTOEXPOSURELIGHTINGMODE
 - CameraDefsC.h, [244](#)
- NUM_AUTOEXPOSUREMETERINGMODE
 - CameraDefsC.h, [245](#)
- NUM_AUTOEXPOSURETARGETGREYVALUEAUTO
 - CameraDefsC.h, [245](#)
- NUM_BALANCERATIOSELECTOR
 - CameraDefsC.h, [245](#)
- NUM_BALANCEWHITEAUTO
 - CameraDefsC.h, [246](#)
- NUM_BALANCEWHITEAUTOPROFILE
 - CameraDefsC.h, [246](#)
- NUM_BINNINGHORIZONTALMODE
 - CameraDefsC.h, [246](#)
- NUM_BINNINGSELECTOR
 - CameraDefsC.h, [247](#)
- NUM_BINNINGVERTICALMODE
 - CameraDefsC.h, [247](#)
- NUM_BLACKLEVELAUTO
 - CameraDefsC.h, [248](#)
- NUM_BLACKLEVELAUTOBALANCE
 - CameraDefsC.h, [247](#)
- NUM_BLACKLEVELSELECTOR
 - CameraDefsC.h, [248](#)
- NUM_CHUNKBLACKLEVELSELECTOR
 - CameraDefsC.h, [248](#)
- NUM_CHUNKCOUNTERSELECTOR
 - CameraDefsC.h, [249](#)
- NUM_CHUNKENCODERSELECTOR
 - CameraDefsC.h, [249](#)
- NUM_CHUNKENCODERSTATUS
 - CameraDefsC.h, [249](#)
- NUM_CHUNKEXPOSURETIMESELECTOR
 - CameraDefsC.h, [250](#)
- NUM_CHUNKGAINSELECTOR
 - CameraDefsC.h, [250](#)
- NUM_CHUNKIMAGECOMPONENT
 - CameraDefsC.h, [251](#)
- NUM_CHUNKPIXELFORMAT
 - CameraDefsC.h, [251](#)
- NUM_CHUNKREGIONID
 - CameraDefsC.h, [251](#)
- NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR
 - CameraDefsC.h, [252](#)
- NUM_CHUNKSCAN3DCOORDINATESELECTOR
 - CameraDefsC.h, [252](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEM
 - CameraDefsC.h, [252](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE
 - CameraDefsC.h, [253](#)
- NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR
 - CameraDefsC.h, [253](#)
- NUM_CHUNKSCAN3DDISTANCEUNIT
 - CameraDefsC.h, [253](#)
- NUM_CHUNKSCAN3DOUTPUTMODE
 - CameraDefsC.h, [254](#)
- NUM_CHUNKSELECTOR
 - CameraDefsC.h, [255](#)
- NUM_CHUNKSOURCEID
 - CameraDefsC.h, [255](#)
- NUM_CHUNKTIMERSELECTOR
 - CameraDefsC.h, [256](#)
- NUM_CHUNKTRANSFERSTREAMID
 - CameraDefsC.h, [256](#)
- NUM_CLCONFIGURATION
 - CameraDefsC.h, [256](#)

- NUM_CLTIMESLOTSCOUNT
 - CameraDefsC.h, [257](#)
- NUM_COLORTRANSFORMATIONSELECTOR
 - CameraDefsC.h, [257](#)
- NUM_COLORTRANSFORMATIONVALUESELECTOR
 - CameraDefsC.h, [257](#)
- NUM_COMPRESSIONSATURATIONPRIORITY
 - CameraDefsC.h, [258](#)
- NUM_COUNTEREVENTACTIVATION
 - CameraDefsC.h, [258](#)
- NUM_COUNTEREVENTSOURCE
 - CameraDefsC.h, [259](#)
- NUM_COUNTERRESETACTIVATION
 - CameraDefsC.h, [259](#)
- NUM_COUNTERRESETSOURCE
 - CameraDefsC.h, [260](#)
- NUM_COUNTERSELECTOR
 - CameraDefsC.h, [260](#)
- NUM_COUNTERSTATUS
 - CameraDefsC.h, [260](#)
- NUM_COUNTERTRIGGERACTIVATION
 - CameraDefsC.h, [261](#)
- NUM_COUNTERTRIGGERSOURCE
 - CameraDefsC.h, [261](#)
- NUM_CXPCONNECTIONTESTMODE
 - CameraDefsC.h, [262](#)
- NUM_CXPLINKCONFIGURATION
 - CameraDefsC.h, [263](#)
- NUM_CXPLINKCONFIGURATIONPREFERRED
 - CameraDefsC.h, [264](#)
- NUM_CXPLINKCONFIGURATIONSTATUS
 - CameraDefsC.h, [265](#)
- NUM_CXPPOCXPSTATUS
 - CameraDefsC.h, [265](#)
- NUM_DECIMATIONHORIZONTALMODE
 - CameraDefsC.h, [265](#)
- NUM_DECIMATIONSELECTOR
 - CameraDefsC.h, [266](#)
- NUM_DECIMATIONVERTICALMODE
 - CameraDefsC.h, [266](#)
- NUM_DEFECTCORRECTIONMODE
 - CameraDefsC.h, [266](#)
- NUM_DEINTERLACING
 - CameraDefsC.h, [267](#)
- NUM_DEVICECHARACTERSET
 - CameraDefsC.h, [267](#)
- NUM_DEVICECLOCKSELECTOR
 - CameraDefsC.h, [267](#)
- NUM_DEVICECONNECTIONSTATUS
 - CameraDefsC.h, [267](#)
- NUM_DEVICEINDICATORMODE
 - CameraDefsC.h, [268](#)
- NUM_DEVICELINKHEARTBEATMODE
 - CameraDefsC.h, [268](#)
- NUM_DEVICELINKTHROUGHPUTLIMITMODE
 - CameraDefsC.h, [268](#)
- NUM_DEVICEPOWERSUPPLYSELECTOR
 - CameraDefsC.h, [269](#)
- NUM_DEVICEREGISTERSENDIANNESS
 - CameraDefsC.h, [269](#)
- NUM_DEVICESCANTYPE
 - CameraDefsC.h, [269](#)
- NUM_DEVICESERIALPORTBAUDRATE
 - CameraDefsC.h, [270](#)
- NUM_DEVICESERIALPORTSELECTOR
 - CameraDefsC.h, [270](#)
- NUM_DEVICESTREAMCHANNELENDIANNESS
 - CameraDefsC.h, [270](#)
- NUM_DEVICESTREAMCHANNELTYPE
 - CameraDefsC.h, [271](#)
- NUM_DEVICETAPGEOMETRY
 - CameraDefsC.h, [272](#)
- NUM_DEVICETEMPERATURESELECTOR
 - CameraDefsC.h, [272](#)
- NUM_DEVICETLTYPE
 - CameraDefsC.h, [272](#)
- NUM_DEVICETYPE
 - CameraDefsC.h, [273](#)
- NUM_ENCODERMODE
 - CameraDefsC.h, [273](#)
- NUM_ENCODEROUTPUTMODE
 - CameraDefsC.h, [274](#)
- NUM_ENCODERRESETACTIVATION
 - CameraDefsC.h, [274](#)
- NUM_ENCODERRESETSOURCE
 - CameraDefsC.h, [275](#)
- NUM_ENCODERSELECTOR
 - CameraDefsC.h, [275](#)
- NUM_ENCODERSOURCEA
 - CameraDefsC.h, [276](#)
- NUM_ENCODERSOURCEB
 - CameraDefsC.h, [276](#)
- NUM_ENCODERSTATUS
 - CameraDefsC.h, [276](#)
- NUM_EVENTNOTIFICATION
 - CameraDefsC.h, [277](#)
- NUM_EVENTSELECTOR
 - CameraDefsC.h, [277](#)
- NUM_EXPOSUREACTIVEMODE
 - CameraDefsC.h, [277](#)
- NUM_EXPOSUREAUTO
 - CameraDefsC.h, [278](#)
- NUM_EXPOSUREMODE
 - CameraDefsC.h, [278](#)
- NUM_EXPOSURETIMEMODE
 - CameraDefsC.h, [278](#)
- NUM_EXPOSURETIMESELECTOR
 - CameraDefsC.h, [279](#)
- NUM_FILEOPENMODE
 - CameraDefsC.h, [279](#)
- NUM_FILEOPERATIONSELECTOR
 - CameraDefsC.h, [280](#)
- NUM_FILEOPERATIONSTATUS
 - CameraDefsC.h, [280](#)
- NUM_FILESELECTOR
 - CameraDefsC.h, [280](#)

- NUM_GAINAUTO
CameraDefsC.h, [282](#)
- NUM_GAINAUTOBALANCE
CameraDefsC.h, [282](#)
- NUM_GAINSELECTOR
CameraDefsC.h, [282](#)
- NUM_GEVCCP
CameraDefsC.h, [283](#)
- NUM_GEVCURRENTPHYSICALLINKCONFIGURATION
CameraDefsC.h, [283](#)
- NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR
CameraDefsC.h, [283](#)
- NUM_GEVGVSPEXTENDEDIDMODE
CameraDefsC.h, [284](#)
- NUM_GEVEEEE1588CLOCKACCURACY
CameraDefsC.h, [284](#)
- NUM_GEVEEEE1588MODE
CameraDefsC.h, [284](#)
- NUM_GEVEEEE1588STATUS
CameraDefsC.h, [285](#)
- NUM_GEVIPCONFIGURATIONSTATUS
CameraDefsC.h, [285](#)
- NUM_GEVPHYSCALLINKCONFIGURATION
CameraDefsC.h, [285](#)
- NUM_GEVSUPPORTEDOPTIONSELECTOR
CameraDefsC.h, [286](#)
- NUM_IMAGECOMPONENTSELECTOR
CameraDefsC.h, [287](#)
- NUM_IMAGECOMPRESSIONJPEGFORMATOPTION
CameraDefsC.h, [287](#)
- NUM_IMAGECOMPRESSIONMODE
CameraDefsC.h, [288](#)
- NUM_IMAGECOMPRESSIONRATEOPTION
CameraDefsC.h, [288](#)
- NUM_LINEFORMAT
CameraDefsC.h, [288](#)
- NUM_LINEINPUTFILTERSELECTOR
CameraDefsC.h, [289](#)
- NUM_LINEMODE
CameraDefsC.h, [289](#)
- NUM_LINESELECTOR
CameraDefsC.h, [289](#)
- NUM_LINESOURCE
CameraDefsC.h, [290](#)
- NUM_LOGICBLOCKLUTINPUTACTIVATION
CameraDefsC.h, [290](#)
- NUM_LOGICBLOCKLUTINPUTSELECTOR
CameraDefsC.h, [291](#)
- NUM_LOGICBLOCKLUTINPUTSOURCE
CameraDefsC.h, [291](#)
- NUM_LOGICBLOCKLUTSELECTOR
CameraDefsC.h, [292](#)
- NUM_LOGICBLOCKSELECTOR
CameraDefsC.h, [292](#)
- NUM_LUTSELECTOR
CameraDefsC.h, [292](#)
- NUM_PIXELCOLORFILTER
CameraDefsC.h, [293](#)
- NUM_PIXELFORMAT
CameraDefsC.h, [298](#)
- NUM_PIXELFORMATINFOSELECTOR
CameraDefsC.h, [304](#)
- NUM_PIXELSIZE
CameraDefsC.h, [305](#)
- NUM_REGIONDESTINATION
CameraDefsC.h, [305](#)
- NUM_REGIONMODE
CameraDefsC.h, [305](#)
- NUM_REGIONSELECTOR
CameraDefsC.h, [306](#)
- NUM_RGBTRANSFORMLIGHTSOURCE
CameraDefsC.h, [306](#)
- NUM_SCAN3DCOORDINATEREFERENCESELECTOR
CameraDefsC.h, [307](#)
- NUM_SCAN3DCOORDINATESELECTOR
CameraDefsC.h, [307](#)
- NUM_SCAN3DCOORDINATESYSTEM
CameraDefsC.h, [307](#)
- NUM_SCAN3DCOORDINATESYSTEMREFERENCE
CameraDefsC.h, [308](#)
- NUM_SCAN3DCOORDINATETRANSFORMSELECTOR
CameraDefsC.h, [308](#)
- NUM_SCAN3DDISTANCEUNIT
CameraDefsC.h, [308](#)
- NUM_SCAN3DOUTPUTMODE
CameraDefsC.h, [309](#)
- NUM_SENSORDIGITIZATIONTAPS
CameraDefsC.h, [310](#)
- NUM_SENSORSHUTTERMODE
CameraDefsC.h, [310](#)
- NUM_SENSORTAPS
CameraDefsC.h, [310](#)
- NUM_SEQUENCERCONFIGURATIONMODE
CameraDefsC.h, [311](#)
- NUM_SEQUENCERCONFIGURATIONVALID
CameraDefsC.h, [311](#)
- NUM_SEQUENCERMODE
CameraDefsC.h, [311](#)
- NUM_SEQUENCERSETVALID
CameraDefsC.h, [312](#)
- NUM_SEQUENCERTRIGGERACTIVATION
CameraDefsC.h, [312](#)
- NUM_SEQUENCERTRIGGERSOURCE
CameraDefsC.h, [312](#)
- NUM_SERIALPORTBAUDRATE
CameraDefsC.h, [313](#)
- NUM_SERIALPORTPARITY
CameraDefsC.h, [313](#)
- NUM_SERIALPORTSELECTOR
CameraDefsC.h, [313](#)
- NUM_SERIALPORTSOURCE
CameraDefsC.h, [314](#)
- NUM_SERIALPORTSTOPBITS
CameraDefsC.h, [314](#)
- NUM_SOFTWARESIGNALSELECTOR
CameraDefsC.h, [314](#)

- NUM_SOURCESELECTOR
 - CameraDefsC.h, [315](#)
- NUM_TESTPATTERN
 - CameraDefsC.h, [315](#)
- NUM_TESTPATTERNGENERATORSELECTOR
 - CameraDefsC.h, [315](#)
- NUM_TIMERSELECTOR
 - CameraDefsC.h, [316](#)
- NUM_TIMERSTATUS
 - CameraDefsC.h, [316](#)
- NUM_TIMERTRIGGERACTIVATION
 - CameraDefsC.h, [316](#)
- NUM_TIMERTRIGGERSOURCE
 - CameraDefsC.h, [318](#)
- NUM_TRANSFERCOMPONENTSELECTOR
 - CameraDefsC.h, [318](#)
- NUM_TRANSFERCONTROLMODE
 - CameraDefsC.h, [319](#)
- NUM_TRANSFEROPERATIONMODE
 - CameraDefsC.h, [319](#)
- NUM_TRANSFERQUEUEMODE
 - CameraDefsC.h, [319](#)
- NUM_TRANSFERSELECTOR
 - CameraDefsC.h, [319](#)
- NUM_TRANSFERSTATUSSELECTOR
 - CameraDefsC.h, [320](#)
- NUM_TRANSFERTRIGGERACTIVATION
 - CameraDefsC.h, [320](#)
- NUM_TRANSFERTRIGGERMODE
 - CameraDefsC.h, [321](#)
- NUM_TRANSFERTRIGGERSELECTOR
 - CameraDefsC.h, [321](#)
- NUM_TRANSFERTRIGGERSOURCE
 - CameraDefsC.h, [322](#)
- NUM_TRIGGERACTIVATION
 - CameraDefsC.h, [322](#)
- NUM_TRIGGERMODE
 - CameraDefsC.h, [323](#)
- NUM_TRIGGEROVERLAP
 - CameraDefsC.h, [323](#)
- NUM_TRIGGERSELECTOR
 - CameraDefsC.h, [323](#)
- NUM_TRIGGERSOURCE
 - CameraDefsC.h, [324](#)
- NUM_USEROUTPUTSELECTOR
 - CameraDefsC.h, [324](#)
- NUM_USERSETDEFAULT
 - CameraDefsC.h, [325](#)
- NUM_USERSETSELECTOR
 - CameraDefsC.h, [325](#)
- NUM_WHITECLIPSELECTOR
 - CameraDefsC.h, [325](#)
- NUMDEVICEACCESSSTATUS
 - TransportLayerDefsC.h, [509](#)
- NUMDEVICECURRENTSPEED
 - TransportLayerDefsC.h, [509](#)
- NUMDEVICEENDIANESSMECHANISM
 - TransportLayerDefsC.h, [511](#)
- NUMDEVICETYPE
 - TransportLayerDefsC.h, [511](#)
- NUMFILTERDRIVERSTATUS
 - TransportLayerDefsC.h, [511](#)
- NUMGENICAMXMLLOCATION
 - TransportLayerDefsC.h, [512](#)
- NUMGEVCCP
 - TransportLayerDefsC.h, [512](#)
- NUMGUIXMLLOCATION
 - TransportLayerDefsC.h, [512](#)
- NUMINTERFACETYPE
 - TransportLayerDefsC.h, [513](#)
- NUMPOESTATUS
 - TransportLayerDefsC.h, [513](#)
- NUMSTREAMBUFFERCOUNTMODE
 - TransportLayerDefsC.h, [513](#)
- NUMSTREAMBUFFERHANDLINGMODE
 - TransportLayerDefsC.h, [514](#)
- NUMSTREAMMODE
 - TransportLayerDefsC.h, [514](#)
- NUMSTREAMTYPE
 - TransportLayerDefsC.h, [515](#)
- NUMTLTYPE
 - TransportLayerDefsC.h, [515](#)
- OffsetX
 - _quickSpin, [149](#)
- OffsetY
 - _quickSpin, [149](#)
- PacketResendRequestCount
 - _quickSpin, [149](#)
- PayloadSize
 - _quickSpin, [149](#)
- PixelColorFilter
 - _quickSpin, [149](#)
- PixelColorFilter_BayerBG
 - CameraDefsC.h, [293](#)
- PixelColorFilter_BayerGB
 - CameraDefsC.h, [293](#)
- PixelColorFilter_BayerGR
 - CameraDefsC.h, [293](#)
- PixelColorFilter_BayerRG
 - CameraDefsC.h, [292](#)
- PixelColorFilter_None
 - CameraDefsC.h, [292](#)
- PixelDynamicRangeMax
 - _quickSpin, [149](#)
- PixelDynamicRangeMin
 - _quickSpin, [149](#)
- PixelFormat
 - _quickSpin, [149](#)
- PixelFormat_B10
 - CameraDefsC.h, [295](#)
- PixelFormat_B12
 - CameraDefsC.h, [295](#)
- PixelFormat_B12_Jpeg
 - CameraDefsC.h, [298](#)
- PixelFormat_B16

- CameraDefsC.h, [295](#)
- PixelFormat_B8
 - CameraDefsC.h, [295](#)
- PixelFormat_BayerBG10
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerBG10p
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerBG10Packed
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerBG12
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerBG12p
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerBG12Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerBG16
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerBG8
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGB10
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerGB10p
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerGB10Packed
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerGB12
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerGB12p
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGB12Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGB16
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGB8
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGR10
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerGR10p
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerGR10Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGR12
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerGR12p
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGR12Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGR16
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerGR8
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerRG10
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerRG10p
 - CameraDefsC.h, [294](#)
- PixelFormat_BayerRG10Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerRG12
 - CameraDefsC.h, [294](#)
- CameraDefsC.h, [294](#)
- PixelFormat_BayerRG12p
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerRG12Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerRG16
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerRG8
 - CameraDefsC.h, [293](#)
- PixelFormat_BayerRGPolarized10p
 - CameraDefsC.h, [298](#)
- PixelFormat_BayerRGPolarized12p
 - CameraDefsC.h, [298](#)
- PixelFormat_BayerRGPolarized16
 - CameraDefsC.h, [298](#)
- PixelFormat_BayerRGPolarized8
 - CameraDefsC.h, [298](#)
- PixelFormat_BGR10
 - CameraDefsC.h, [295](#)
- PixelFormat_BGR10p
 - CameraDefsC.h, [295](#)
- PixelFormat_BGR12
 - CameraDefsC.h, [295](#)
- PixelFormat_BGR12p
 - CameraDefsC.h, [295](#)
- PixelFormat_BGR14
 - CameraDefsC.h, [295](#)
- PixelFormat_BGR16
 - CameraDefsC.h, [295](#)
- PixelFormat_BGR565p
 - CameraDefsC.h, [295](#)
- PixelFormat_BGR8
 - CameraDefsC.h, [293](#)
- PixelFormat_BGRa10
 - CameraDefsC.h, [295](#)
- PixelFormat_BGRa10p
 - CameraDefsC.h, [295](#)
- PixelFormat_BGRa12
 - CameraDefsC.h, [295](#)
- PixelFormat_BGRa12p
 - CameraDefsC.h, [295](#)
- PixelFormat_BGRa14
 - CameraDefsC.h, [295](#)
- PixelFormat_BGRa16
 - CameraDefsC.h, [295](#)
- PixelFormat_BGRa8
 - CameraDefsC.h, [293](#)
- PixelFormat_BiColorBGRG10
 - CameraDefsC.h, [296](#)
- PixelFormat_BiColorBGRG10p
 - CameraDefsC.h, [296](#)
- PixelFormat_BiColorBGRG12
 - CameraDefsC.h, [296](#)
- PixelFormat_BiColorBGRG12p
 - CameraDefsC.h, [296](#)
- PixelFormat_BiColorBGRG8
 - CameraDefsC.h, [296](#)
- PixelFormat_BiColorRGBG10

- CameraDefsC.h, [296](#)
- PixelFormat_BiColorRGBG10p
 - CameraDefsC.h, [296](#)
- PixelFormat_BiColorRGBG12
 - CameraDefsC.h, [296](#)
- PixelFormat_BiColorRGBG12p
 - CameraDefsC.h, [296](#)
- PixelFormat_BiColorRGBG8
 - CameraDefsC.h, [296](#)
- PixelFormat_Confidence1
 - CameraDefsC.h, [296](#)
- PixelFormat_Confidence16
 - CameraDefsC.h, [296](#)
- PixelFormat_Confidence1p
 - CameraDefsC.h, [296](#)
- PixelFormat_Confidence32f
 - CameraDefsC.h, [296](#)
- PixelFormat_Confidence8
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_A10p
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_A12p
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_A16
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_A32f
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_A8
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_ABC10p
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC10p_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC12p
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC12p_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC16
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC16_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC32f
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC32f_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC8
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_ABC8_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC10p
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC10p_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC12p
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC12p_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC16
 - CameraDefsC.h, [295](#)
- CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC16_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC32f
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC32f_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC8
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_AC8_Planar
 - CameraDefsC.h, [295](#)
- PixelFormat_Coord3D_B10p
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_B12p
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_B16
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_B32f
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_B8
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_C10p
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_C12p
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_C16
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_C32f
 - CameraDefsC.h, [296](#)
- PixelFormat_Coord3D_C8
 - CameraDefsC.h, [296](#)
- PixelFormat_G10
 - CameraDefsC.h, [295](#)
- PixelFormat_G12
 - CameraDefsC.h, [295](#)
- PixelFormat_G16
 - CameraDefsC.h, [295](#)
- PixelFormat_G8
 - CameraDefsC.h, [295](#)
- PixelFormat_GB12
 - CameraDefsC.h, [298](#)
- PixelFormat_GB12_Jpeg
 - CameraDefsC.h, [298](#)
- PixelFormat_GR12
 - CameraDefsC.h, [298](#)
- PixelFormat_GR12_Jpeg
 - CameraDefsC.h, [298](#)
- PixelFormat_JPEGColor8
 - CameraDefsC.h, [298](#)
- PixelFormat_JPEGMono8
 - CameraDefsC.h, [298](#)
- PixelFormat_LLCBayerRG8
 - CameraDefsC.h, [298](#)
- PixelFormat_LLCMono8
 - CameraDefsC.h, [298](#)
- PixelFormat_Mono10
 - CameraDefsC.h, [294](#)
- PixelFormat_Mono10p

- CameraDefsC.h, [294](#)
- PixelFormat_Mono10Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_Mono12
 - CameraDefsC.h, [294](#)
- PixelFormat_Mono12p
 - CameraDefsC.h, [293](#)
- PixelFormat_Mono12Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_Mono14
 - CameraDefsC.h, [294](#)
- PixelFormat_Mono16
 - CameraDefsC.h, [293](#)
- PixelFormat_Mono16s
 - CameraDefsC.h, [294](#)
- PixelFormat_Mono1p
 - CameraDefsC.h, [294](#)
- PixelFormat_Mono2p
 - CameraDefsC.h, [294](#)
- PixelFormat_Mono32f
 - CameraDefsC.h, [294](#)
- PixelFormat_Mono4p
 - CameraDefsC.h, [294](#)
- PixelFormat_Mono8
 - CameraDefsC.h, [293](#)
- PixelFormat_Mono8s
 - CameraDefsC.h, [294](#)
- PixelFormat_Polarized10p
 - CameraDefsC.h, [298](#)
- PixelFormat_Polarized12p
 - CameraDefsC.h, [298](#)
- PixelFormat_Polarized16
 - CameraDefsC.h, [298](#)
- PixelFormat_Polarized8
 - CameraDefsC.h, [298](#)
- PixelFormat_R10
 - CameraDefsC.h, [295](#)
- PixelFormat_R12
 - CameraDefsC.h, [295](#)
- PixelFormat_R12_Jpeg
 - CameraDefsC.h, [298](#)
- PixelFormat_R16
 - CameraDefsC.h, [295](#)
- PixelFormat_R8
 - CameraDefsC.h, [295](#)
- PixelFormat_Raw16
 - CameraDefsC.h, [298](#)
- PixelFormat_Raw8
 - CameraDefsC.h, [298](#)
- PixelFormat_RGB10
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB10_Planar
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB10p
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB10p32
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB12
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB12_Planar
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB12p
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB14
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB16
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB16_Planar
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB16s
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB32f
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB565p
 - CameraDefsC.h, [295](#)
- PixelFormat_RGB8
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB8_Planar
 - CameraDefsC.h, [294](#)
- PixelFormat_RGB8Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_RGBa10
 - CameraDefsC.h, [294](#)
- PixelFormat_RGBa10p
 - CameraDefsC.h, [294](#)
- PixelFormat_RGBa12
 - CameraDefsC.h, [294](#)
- PixelFormat_RGBa12p
 - CameraDefsC.h, [294](#)
- PixelFormat_RGBa14
 - CameraDefsC.h, [294](#)
- PixelFormat_RGBa16
 - CameraDefsC.h, [294](#)
- PixelFormat_RGBa32f
 - CameraDefsC.h, [295](#)
- PixelFormat_RGBa8
 - CameraDefsC.h, [294](#)
- PixelFormat_SCF1WBWG10
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WBWG10p
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WBWG12
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WBWG12p
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WBWG14
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WBWG16
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WBWG8
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WGWB10
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WGWB10p
 - CameraDefsC.h, [296](#)
- PixelFormat_SCF1WGWB12

CameraDefsC.h, [296](#)
PixelFormat_SCF1WGW B12p
CameraDefsC.h, [296](#)
PixelFormat_SCF1WGW B14
CameraDefsC.h, [296](#)
PixelFormat_SCF1WGW B16
CameraDefsC.h, [296](#)
PixelFormat_SCF1WGW B8
CameraDefsC.h, [296](#)
PixelFormat_SCF1WGW R10
CameraDefsC.h, [296](#)
PixelFormat_SCF1WGW R10p
CameraDefsC.h, [296](#)
PixelFormat_SCF1WGW R12
CameraDefsC.h, [297](#)
PixelFormat_SCF1WGW R12p
CameraDefsC.h, [297](#)
PixelFormat_SCF1WGW R14
CameraDefsC.h, [297](#)
PixelFormat_SCF1WGW R16
CameraDefsC.h, [297](#)
PixelFormat_SCF1WGW R8
CameraDefsC.h, [296](#)
PixelFormat_SCF1WRW G10
CameraDefsC.h, [297](#)
PixelFormat_SCF1WRW G10p
CameraDefsC.h, [297](#)
PixelFormat_SCF1WRW G12
CameraDefsC.h, [297](#)
PixelFormat_SCF1WRW G12p
CameraDefsC.h, [297](#)
PixelFormat_SCF1WRW G14
CameraDefsC.h, [297](#)
PixelFormat_SCF1WRW G16
CameraDefsC.h, [297](#)
PixelFormat_SCF1WRW G8
CameraDefsC.h, [297](#)
PixelFormat_YCbCr10_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr10p_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr12_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr12p_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr411_8
CameraDefsC.h, [293](#)
PixelFormat_YCbCr411_8_CbYYCrYY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_10
CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_10_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_10p
CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_10p_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_12

CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_12_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_12p
CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_12p_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr422_8
CameraDefsC.h, [293](#)
PixelFormat_YCbCr422_8_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_10_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_10p_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_12_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_12p_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_411_8_CbYYCrYY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_10
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_10_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_10p
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_10p_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_12
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_12_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_12p
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_12p_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_8
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_422_8_CbYCrY
CameraDefsC.h, [297](#)
PixelFormat_YCbCr601_8_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr709_10_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr709_10p_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr709_12_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr709_12p_CbYCr
CameraDefsC.h, [297](#)
PixelFormat_YCbCr709_411_8_CbYYCrYY
CameraDefsC.h, [298](#)
PixelFormat_YCbCr709_422_10
CameraDefsC.h, [298](#)
PixelFormat_YCbCr709_422_10_CbYCrY
CameraDefsC.h, [298](#)
PixelFormat_YCbCr709_422_10p

- CameraDefsC.h, [298](#)
- PixelFormat_YCbCr709_422_10p_CbYCrY
 - CameraDefsC.h, [298](#)
- PixelFormat_YCbCr709_422_12
 - CameraDefsC.h, [298](#)
- PixelFormat_YCbCr709_422_12_CbYCrY
 - CameraDefsC.h, [298](#)
- PixelFormat_YCbCr709_422_12p
 - CameraDefsC.h, [298](#)
- PixelFormat_YCbCr709_422_12p_CbYCrY
 - CameraDefsC.h, [298](#)
- PixelFormat_YCbCr709_422_8
 - CameraDefsC.h, [298](#)
- PixelFormat_YCbCr709_422_8_CbYCrY
 - CameraDefsC.h, [298](#)
- PixelFormat_YCbCr709_8_CbYCr
 - CameraDefsC.h, [297](#)
- PixelFormat_YCbCr8
 - CameraDefsC.h, [293](#)
- PixelFormat_YCbCr8_CbYCr
 - CameraDefsC.h, [297](#)
- PixelFormat_YUV411_8_UYYVYY
 - CameraDefsC.h, [298](#)
- PixelFormat_YUV411Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_YUV422_8
 - CameraDefsC.h, [298](#)
- PixelFormat_YUV422_8_UYVY
 - CameraDefsC.h, [298](#)
- PixelFormat_YUV422Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_YUV444Packed
 - CameraDefsC.h, [293](#)
- PixelFormat_YUV8_UYV
 - CameraDefsC.h, [298](#)
- PixelFormatInfoID
 - _quickSpin, [150](#)
- PixelFormatInfoSelector
 - _quickSpin, [150](#)
- PixelFormatInfoSelector_B10
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_B12
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_B16
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_B8
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BayerBG10
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerBG10p
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerBG12
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerBG12p
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerBG16
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerBG8
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGB10
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGB10p
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGB12
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGB12p
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGB16
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGB8
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGR10
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGR10p
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGR12
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGR12p
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGR16
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerGR8
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerRG10
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerRG10p
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerRG12
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerRG12p
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerRG16
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerRG8
 - CameraDefsC.h, [299](#)
- PixelFormatInfoSelector_BayerRGPolarized10p
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_BayerRGPolarized12p
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_BayerRGPolarized16
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_BayerRGPolarized8
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_BGR10
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGR10p
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGR12
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGR12p
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGR14
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGR16
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGR565p

- CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGR8
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGRa10
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGRa10p
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGRa12
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGRa12p
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGRa14
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGRa16
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BGRa8
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_BiColorBGRG10
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorBGRG10p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorBGRG12
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorBGRG12p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorBGRG8
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorRGBG10
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorRGBG10p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorRGBG12
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorRGBG12p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_BiColorRGBG8
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Confidence1
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Confidence16
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Confidence1p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Confidence32f
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Confidence8
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_A10p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_A12p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_A16
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_A32f
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_A8
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC10p
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC10p_Planar
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC12p
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC12p_Planar
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC16
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC16_Planar
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC32f
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC32f_Planar
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_ABC8
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_Coord3D_ABC8_Planar
 - CameraDefsC.h, [300](#)
- PixelFormatInfoSelector_Coord3D_AC10p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC10p_Planar
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC12p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC12p_Planar
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC16
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC16_Planar
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC32f
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC32f_Planar
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC8
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_AC8_Planar
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_B10p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_B12p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_B16
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_B32f
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_B8
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_C10p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_C12p
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_C16
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_C32f
 - CameraDefsC.h, [301](#)
- PixelFormatInfoSelector_Coord3D_C8
 - CameraDefsC.h, [301](#)

CameraDefsC.h, [301](#)
 PixelFormatInfoSelector_G10
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_G12
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_G16
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_G8
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_JPEGColor8
 CameraDefsC.h, [304](#)
 PixelFormatInfoSelector_JPEGMono8
 CameraDefsC.h, [304](#)
 PixelFormatInfoSelector_LLCBayerRG8
 CameraDefsC.h, [304](#)
 PixelFormatInfoSelector_LLCMono8
 CameraDefsC.h, [304](#)
 PixelFormatInfoSelector_Mono10
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono10p
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono12
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono12p
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono14
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono16
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono16s
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono1p
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono2p
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono32f
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono4p
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono8
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Mono8s
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_Polarized10p
 CameraDefsC.h, [304](#)
 PixelFormatInfoSelector_Polarized12p
 CameraDefsC.h, [304](#)
 PixelFormatInfoSelector_Polarized16
 CameraDefsC.h, [304](#)
 PixelFormatInfoSelector_Polarized8
 CameraDefsC.h, [304](#)
 PixelFormatInfoSelector_R10
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_R12
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_R16
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_R8

CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB10
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGB10_Planar
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB10p
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB10p32
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB12
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB12_Planar
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB12p
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB14
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB16
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB16_Planar
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB16s
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB32f
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB565p
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGB8
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGB8_Planar
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGBa10
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGBa10p
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGBa12
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGBa12p
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGBa14
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGBa16
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_RGBa32f
 CameraDefsC.h, [300](#)
 PixelFormatInfoSelector_RGBa8
 CameraDefsC.h, [299](#)
 PixelFormatInfoSelector_SCF1WBWG10
 CameraDefsC.h, [301](#)
 PixelFormatInfoSelector_SCF1WBWG10p
 CameraDefsC.h, [302](#)
 PixelFormatInfoSelector_SCF1WBWG12
 CameraDefsC.h, [302](#)
 PixelFormatInfoSelector_SCF1WBWG12p
 CameraDefsC.h, [302](#)
 PixelFormatInfoSelector_SCF1WBWG14
 CameraDefsC.h, [302](#)
 PixelFormatInfoSelector_SCF1WBWG16

- CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WBWG8
CameraDefsC.h, [301](#)
PixelFormatInfoSelector_SCF1WGWB10
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWB10p
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWB12
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWB12p
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWB14
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWB16
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWB8
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWR10
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWR10p
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWR12
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWR12p
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWR14
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWR16
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WGWR8
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WRWG10
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WRWG10p
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WRWG12
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WRWG12p
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WRWG14
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WRWG16
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_SCF1WRWG8
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_YCbCr10_CbYCr
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_YCbCr10p_CbYCr
CameraDefsC.h, [302](#)
PixelFormatInfoSelector_YCbCr12_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr12p_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr411_8
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_10
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_10_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_10p
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_12
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_12_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_12p
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_8
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr422_8_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_10_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_10p_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_12_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_12p_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_10
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_10p
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_12
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_12p
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_8
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr601_8_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr709_10_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr709_10p_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr709_12_CbYCr
CameraDefsC.h, [303](#)
PixelFormatInfoSelector_YCbCr709_12p_CbYCr

- CameraDefsC.h, [303](#)
- PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY
 - CameraDefsC.h, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_10
 - CameraDefsC.h, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY
 - CameraDefsC.h, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_10p
 - CameraDefsC.h, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_12
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_12p
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_YCbCr709_422_8
 - CameraDefsC.h, [303](#)
- PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY
 - CameraDefsC.h, [303](#)
- PixelFormatInfoSelector_YCbCr709_8_CbYCr
 - CameraDefsC.h, [303](#)
- PixelFormatInfoSelector_YCbCr8
 - CameraDefsC.h, [302](#)
- PixelFormatInfoSelector_YCbCr8_CbYCr
 - CameraDefsC.h, [302](#)
- PixelFormatInfoSelector_YUV411_8_UYYVYY
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_YUV422_8
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_YUV422_8_UYVY
 - CameraDefsC.h, [304](#)
- PixelFormatInfoSelector_YUV8_UYV
 - CameraDefsC.h, [304](#)
- PixelSize
 - _quickSpin, [150](#)
- PixelSize_Bpp1
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp10
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp12
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp14
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp16
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp2
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp20
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp24
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp30
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp32
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp36
 - CameraDefsC.h, [305](#)
- PixelSize_Bpp4
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp48
 - CameraDefsC.h, [305](#)
- PixelSize_Bpp64
 - CameraDefsC.h, [305](#)
- PixelSize_Bpp8
 - CameraDefsC.h, [304](#)
- PixelSize_Bpp96
 - CameraDefsC.h, [305](#)
- POEStatus
 - _quickSpinTLInterface, [180](#)
- POEStatus_NotSupported
 - TransportLayerDefsC.h, [513](#)
- POEStatus_PowerOff
 - TransportLayerDefsC.h, [513](#)
- POEStatus_PowerOn
 - TransportLayerDefsC.h, [513](#)
- PortNode
 - SpinnakerGenApiDefsC.h, [501](#)
- PowerSupplyCurrent
 - _quickSpin, [150](#)
- PowerSupplyVoltage
 - _quickSpin, [150](#)
- progressive
 - _spinJPEGOption, [200](#)
- PureNumber
 - SpinnakerGenApiDefsC.h, [501](#)
- quality
 - _spinJPEGOption, [200](#)
 - _spinJPG2Option, [201](#)
 - _spinMJPGOption, [203](#)
- QuickSpin Access, [25](#)
- quickSpinBooleanNode
 - QuickSpinDefsC.h, [329](#)
- QuickSpinC.h
 - quickSpinInit, [327](#)
 - quickSpinInitEx, [327](#)
 - quickSpinTLDeviceInit, [328](#)
 - quickSpinTLInterfaceInit, [328](#)
 - quickSpinTLStreamInit, [328](#)
 - quickSpinTLSystemInit, [328](#)
- quickSpinCommandNode
 - QuickSpinDefsC.h, [330](#)
- QuickSpinDefsC.h
 - quickSpinBooleanNode, [329](#)
 - quickSpinCommandNode, [330](#)
 - quickSpinEnumerationNode, [330](#)
 - quickSpinFloatNode, [330](#)
 - quickSpinIntegerNode, [330](#)
 - quickSpinRegisterNode, [330](#)
 - quickSpinStringNode, [330](#)
- quickSpinEnumerationNode
 - QuickSpinDefsC.h, [330](#)
- quickSpinFloatNode

- QuickSpinDefsC.h, [330](#)
- quickSpinInit
 - QuickSpinC.h, [327](#)
- quickSpinInitEx
 - QuickSpinC.h, [327](#)
- quickSpinIntegerNode
 - QuickSpinDefsC.h, [330](#)
- quickSpinRegisterNode
 - QuickSpinDefsC.h, [330](#)
- quickSpinStringNode
 - QuickSpinDefsC.h, [330](#)
- quickSpinTLDeviceInit
 - QuickSpinC.h, [328](#)
- quickSpinTLInterfacelInit
 - QuickSpinC.h, [328](#)
- quickSpinTLStreamInit
 - QuickSpinC.h, [328](#)
- quickSpinTLSystemInit
 - QuickSpinC.h, [328](#)
- RegionDestination
 - _quickSpin, [150](#)
- RegionDestination_Stream0
 - CameraDefsC.h, [305](#)
- RegionDestination_Stream1
 - CameraDefsC.h, [305](#)
- RegionDestination_Stream2
 - CameraDefsC.h, [305](#)
- RegionMode
 - _quickSpin, [150](#)
- RegionMode_Off
 - CameraDefsC.h, [305](#)
- RegionMode_On
 - CameraDefsC.h, [305](#)
- RegionSelector
 - _quickSpin, [150](#)
- RegionSelector_All
 - CameraDefsC.h, [306](#)
- RegionSelector_Region0
 - CameraDefsC.h, [306](#)
- RegionSelector_Region1
 - CameraDefsC.h, [306](#)
- RegionSelector_Region2
 - CameraDefsC.h, [306](#)
- RegisterNode
 - SpinnakerGenApiDefsC.h, [501](#)
- reserved
 - _spinAVIOption, [190](#)
 - _spinBMPOption, [191](#)
 - _spinH264Option, [199](#)
 - _spinJPEGOption, [200](#)
 - _spinJPG2Option, [201](#)
 - _spinMJPEGOption, [203](#)
 - _spinPGMOption, [204](#)
 - _spinPNGOption, [205](#)
 - _spinPPMOption, [206](#)
 - _spinTIFFOption, [207](#)
- ReverseX
 - _quickSpin, [151](#)
- ReverseY
 - _quickSpin, [151](#)
- RgbTransformLightSource
 - _quickSpin, [151](#)
- RgbTransformLightSource_Cloudy6500K
 - CameraDefsC.h, [306](#)
- RgbTransformLightSource_CoolFluorescent4000K
 - CameraDefsC.h, [306](#)
- RgbTransformLightSource_Custom
 - CameraDefsC.h, [306](#)
- RgbTransformLightSource_Daylight5000K
 - CameraDefsC.h, [306](#)
- RgbTransformLightSource_General
 - CameraDefsC.h, [306](#)
- RgbTransformLightSource_Shade8000K
 - CameraDefsC.h, [306](#)
- RgbTransformLightSource_Tungsten2800K
 - CameraDefsC.h, [306](#)
- RgbTransformLightSource_WarmFluorescent3000K
 - CameraDefsC.h, [306](#)
- RO
 - SpinnakerGenApiDefsC.h, [497](#)
- RW
 - SpinnakerGenApiDefsC.h, [497](#)
- Saturation
 - _quickSpin, [151](#)
- SaturationEnable
 - _quickSpin, [151](#)
- Scan3dAxisMax
 - _quickSpin, [151](#)
- Scan3dAxisMin
 - _quickSpin, [151](#)
- Scan3dCoordinateOffset
 - _quickSpin, [151](#)
- Scan3dCoordinateReferenceSelector
 - _quickSpin, [152](#)
- Scan3dCoordinateReferenceSelector_RotationX
 - CameraDefsC.h, [307](#)
- Scan3dCoordinateReferenceSelector_RotationY
 - CameraDefsC.h, [307](#)
- Scan3dCoordinateReferenceSelector_RotationZ
 - CameraDefsC.h, [307](#)
- Scan3dCoordinateReferenceSelector_TranslationX
 - CameraDefsC.h, [307](#)
- Scan3dCoordinateReferenceSelector_TranslationY
 - CameraDefsC.h, [307](#)
- Scan3dCoordinateReferenceSelector_TranslationZ
 - CameraDefsC.h, [307](#)
- Scan3dCoordinateReferenceValue
 - _quickSpin, [152](#)
- Scan3dCoordinateScale
 - _quickSpin, [152](#)
- Scan3dCoordinateSelector
 - _quickSpin, [152](#)
- Scan3dCoordinateSelector_CoordinateA
 - CameraDefsC.h, [307](#)
- Scan3dCoordinateSelector_CoordinateB
 - CameraDefsC.h, [307](#)

- Scan3dCoordinateSelector_CoordinateC
CameraDefsC.h, [307](#)
- Scan3dCoordinateSystem
_quickSpin, [152](#)
- Scan3dCoordinateSystem_Cartesian
CameraDefsC.h, [307](#)
- Scan3dCoordinateSystem_Cylindrical
CameraDefsC.h, [307](#)
- Scan3dCoordinateSystem_Spherical
CameraDefsC.h, [307](#)
- Scan3dCoordinateSystemReference
_quickSpin, [152](#)
- Scan3dCoordinateSystemReference_Anchor
CameraDefsC.h, [308](#)
- Scan3dCoordinateSystemReference_Transformed
CameraDefsC.h, [308](#)
- Scan3dCoordinateTransformSelector
_quickSpin, [152](#)
- Scan3dCoordinateTransformSelector_RotationX
CameraDefsC.h, [308](#)
- Scan3dCoordinateTransformSelector_RotationY
CameraDefsC.h, [308](#)
- Scan3dCoordinateTransformSelector_RotationZ
CameraDefsC.h, [308](#)
- Scan3dCoordinateTransformSelector_TranslationX
CameraDefsC.h, [308](#)
- Scan3dCoordinateTransformSelector_TranslationY
CameraDefsC.h, [308](#)
- Scan3dCoordinateTransformSelector_TranslationZ
CameraDefsC.h, [308](#)
- Scan3dDistanceUnit
_quickSpin, [152](#)
- Scan3dDistanceUnit_Inch
CameraDefsC.h, [308](#)
- Scan3dDistanceUnit_Millimeter
CameraDefsC.h, [308](#)
- Scan3dInvalidDataFlag
_quickSpin, [153](#)
- Scan3dInvalidDataValue
_quickSpin, [153](#)
- Scan3dOutputMode
_quickSpin, [153](#)
- Scan3dOutputMode_CalibratedABC_Grid
CameraDefsC.h, [309](#)
- Scan3dOutputMode_CalibratedABC_PointCloud
CameraDefsC.h, [309](#)
- Scan3dOutputMode_CalibratedAC
CameraDefsC.h, [309](#)
- Scan3dOutputMode_CalibratedAC_Linescan
CameraDefsC.h, [309](#)
- Scan3dOutputMode_CalibratedC
CameraDefsC.h, [309](#)
- Scan3dOutputMode_CalibratedC_Linescan
CameraDefsC.h, [309](#)
- Scan3dOutputMode_DisparityC
CameraDefsC.h, [309](#)
- Scan3dOutputMode_DisparityC_Linescan
CameraDefsC.h, [309](#)
- Scan3dOutputMode_RectifiedC
CameraDefsC.h, [309](#)
- Scan3dOutputMode_RectifiedC_Linescan
CameraDefsC.h, [309](#)
- Scan3dOutputMode_UncalibratedC
CameraDefsC.h, [309](#)
- Scan3dTransformValue
_quickSpin, [153](#)
- SensorDescription
_quickSpin, [153](#)
- SensorDigitizationTaps
_quickSpin, [153](#)
- SensorDigitizationTaps_Eight
CameraDefsC.h, [310](#)
- SensorDigitizationTaps_Four
CameraDefsC.h, [310](#)
- SensorDigitizationTaps_One
CameraDefsC.h, [309](#)
- SensorDigitizationTaps_Ten
CameraDefsC.h, [310](#)
- SensorDigitizationTaps_Three
CameraDefsC.h, [310](#)
- SensorDigitizationTaps_Two
CameraDefsC.h, [310](#)
- SensorHeight
_quickSpin, [153](#)
- SensorShutterMode
_quickSpin, [153](#)
- SensorShutterMode_Global
CameraDefsC.h, [310](#)
- SensorShutterMode_GlobalReset
CameraDefsC.h, [310](#)
- SensorShutterMode_Rolling
CameraDefsC.h, [310](#)
- SensorTaps
_quickSpin, [154](#)
- SensorTaps_Eight
CameraDefsC.h, [310](#)
- SensorTaps_Four
CameraDefsC.h, [310](#)
- SensorTaps_One
CameraDefsC.h, [310](#)
- SensorTaps_Ten
CameraDefsC.h, [310](#)
- SensorTaps_Three
CameraDefsC.h, [310](#)
- SensorTaps_Two
CameraDefsC.h, [310](#)
- SensorWidth
_quickSpin, [154](#)
- SequencerConfigurationMode
_quickSpin, [154](#)
- SequencerConfigurationMode_Off
CameraDefsC.h, [311](#)
- SequencerConfigurationMode_On
CameraDefsC.h, [311](#)
- SequencerConfigurationValid
_quickSpin, [154](#)

SequencerConfigurationValid_No
 CameraDefsC.h, [311](#)

SequencerConfigurationValid_Yes
 CameraDefsC.h, [311](#)

SequencerFeatureEnable
 _quickSpin, [154](#)

SequencerMode
 _quickSpin, [154](#)

SequencerMode_Off
 CameraDefsC.h, [311](#)

SequencerMode_On
 CameraDefsC.h, [311](#)

SequencerPathSelector
 _quickSpin, [154](#)

SequencerSetActive
 _quickSpin, [154](#)

SequencerSetLoad
 _quickSpin, [155](#)

SequencerSetNext
 _quickSpin, [155](#)

SequencerSetSave
 _quickSpin, [155](#)

SequencerSetSelector
 _quickSpin, [155](#)

SequencerSetStart
 _quickSpin, [155](#)

SequencerSetValid
 _quickSpin, [155](#)

SequencerSetValid_No
 CameraDefsC.h, [312](#)

SequencerSetValid_Yes
 CameraDefsC.h, [312](#)

SequencerTriggerActivation
 _quickSpin, [155](#)

SequencerTriggerActivation_AnyEdge
 CameraDefsC.h, [312](#)

SequencerTriggerActivation_FallingEdge
 CameraDefsC.h, [312](#)

SequencerTriggerActivation_LevelHigh
 CameraDefsC.h, [312](#)

SequencerTriggerActivation_LevelLow
 CameraDefsC.h, [312](#)

SequencerTriggerActivation_RisingEdge
 CameraDefsC.h, [312](#)

SequencerTriggerSource
 _quickSpin, [155](#)

SequencerTriggerSource_FrameStart
 CameraDefsC.h, [312](#)

SequencerTriggerSource_Off
 CameraDefsC.h, [312](#)

SerialPortBaudRate
 _quickSpin, [156](#)

SerialPortBaudRate_Baud115200
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud1200
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud14400
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud19200
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud230400
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud2400
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud300
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud38400
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud460800
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud4800
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud57600
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud600
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud921600
 CameraDefsC.h, [313](#)

SerialPortBaudRate_Baud9600
 CameraDefsC.h, [313](#)

SerialPortDataBits
 _quickSpin, [156](#)

SerialPortParity
 _quickSpin, [156](#)

SerialPortParity_Even
 CameraDefsC.h, [313](#)

SerialPortParity_Mark
 CameraDefsC.h, [313](#)

SerialPortParity_None
 CameraDefsC.h, [313](#)

SerialPortParity_Odd
 CameraDefsC.h, [313](#)

SerialPortParity_Space
 CameraDefsC.h, [313](#)

SerialPortSelector
 _quickSpin, [156](#)

SerialPortSelector_SerialPort0
 CameraDefsC.h, [313](#)

SerialPortSource
 _quickSpin, [156](#)

SerialPortSource_Line0
 CameraDefsC.h, [314](#)

SerialPortSource_Line1
 CameraDefsC.h, [314](#)

SerialPortSource_Line2
 CameraDefsC.h, [314](#)

SerialPortSource_Line3
 CameraDefsC.h, [314](#)

SerialPortSource_Off
 CameraDefsC.h, [314](#)

SerialPortStopBits
 _quickSpin, [156](#)

SerialPortStopBits_Bits1
 CameraDefsC.h, [314](#)

SerialPortStopBits_Bits1AndAHalf
 CameraDefsC.h, [314](#)

- SerialPortStopBits_Bits2
 - CameraDefsC.h, [314](#)
- SerialReceiveFramingErrorCount
 - _quickSpin, [156](#)
- SerialReceiveParityErrorCount
 - _quickSpin, [156](#)
- SerialReceiveQueueClear
 - _quickSpin, [157](#)
- SerialReceiveQueueCurrentCharacterCount
 - _quickSpin, [157](#)
- SerialReceiveQueueMaxCharacterCount
 - _quickSpin, [157](#)
- SerialTransmitQueueCurrentCharacterCount
 - _quickSpin, [157](#)
- SerialTransmitQueueMaxCharacterCount
 - _quickSpin, [157](#)
- Sharpening
 - _quickSpin, [157](#)
- SharpeningAuto
 - _quickSpin, [157](#)
- SharpeningEnable
 - _quickSpin, [157](#)
- SharpeningThreshold
 - _quickSpin, [158](#)
- Signed
 - SpinnakerGenApiDefsC.h, [501](#)
- SoftwareSignalPulse
 - _quickSpin, [158](#)
- SoftwareSignalSelector
 - _quickSpin, [158](#)
- SoftwareSignalSelector_SoftwareSignal0
 - CameraDefsC.h, [314](#)
- SoftwareSignalSelector_SoftwareSignal1
 - CameraDefsC.h, [314](#)
- SoftwareSignalSelector_SoftwareSignal2
 - CameraDefsC.h, [314](#)
- SourceCount
 - _quickSpin, [158](#)
- SourceSelector
 - _quickSpin, [158](#)
- SourceSelector_All
 - CameraDefsC.h, [315](#)
- SourceSelector_Source0
 - CameraDefsC.h, [315](#)
- SourceSelector_Source1
 - CameraDefsC.h, [315](#)
- SourceSelector_Source2
 - CameraDefsC.h, [315](#)
- spinArrivalEventFunction
 - SpinnakerDefsC.h, [441](#)
- spinBooleanGetValue
 - SpinnakerGenApiC.h, [456](#)
- spinBooleanSetValue
 - SpinnakerGenApiC.h, [456](#)
- spinCamera
 - SpinnakerDefsC.h, [441](#)
- spinCameraBeginAcquisition
 - SpinnakerC.h, [340](#)
- spinCameraDelInit
 - SpinnakerC.h, [341](#)
- spinCameraDiscoverMaxPacketSize
 - SpinnakerC.h, [341](#)
- spinCameraEndAcquisition
 - SpinnakerC.h, [342](#)
- spinCameraForceIP
 - SpinnakerC.h, [342](#)
- spinCameraGetAccessMode
 - SpinnakerC.h, [342](#)
- spinCameraGetGuiXml
 - SpinnakerC.h, [343](#)
- spinCameraGetNextImage
 - SpinnakerC.h, [343](#)
- spinCameraGetNextImageEx
 - SpinnakerC.h, [344](#)
- spinCameraGetNextImageSync
 - SpinnakerC.h, [344](#)
- spinCameraGetNodeMap
 - SpinnakerC.h, [345](#)
- spinCameraGetTLDeviceNodeMap
 - SpinnakerC.h, [346](#)
- spinCameraGetTLStreamNodeMap
 - SpinnakerC.h, [346](#)
- spinCameraGetUniqueID
 - SpinnakerC.h, [346](#)
- spinCameraInit
 - SpinnakerC.h, [347](#)
- spinCameraIsInitialized
 - SpinnakerC.h, [347](#)
- spinCameraIsStreaming
 - SpinnakerC.h, [348](#)
- spinCameraIsValid
 - SpinnakerC.h, [348](#)
- spinCameraList
 - SpinnakerDefsC.h, [441](#)
- spinCameraListAppend
 - SpinnakerC.h, [349](#)
- spinCameraListClear
 - SpinnakerC.h, [349](#)
- spinCameraListCreateEmpty
 - SpinnakerC.h, [350](#)
- spinCameraListDestroy
 - SpinnakerC.h, [350](#)
- spinCameraListGet
 - SpinnakerC.h, [351](#)
- spinCameraListGetBySerial
 - SpinnakerC.h, [351](#)
- spinCameraListGetSize
 - SpinnakerC.h, [352](#)
- spinCameraListRemove
 - SpinnakerC.h, [352](#)
- spinCameraListRemoveBySerial
 - SpinnakerC.h, [353](#)
- spinCameraReadPort
 - SpinnakerC.h, [353](#)
- spinCameraRegisterDeviceEventHandler
 - SpinnakerC.h, [353](#)

- spinCameraRegisterDeviceEventHandlerEx
SpinnakerC.h, [354](#)
- spinCameraRegisterImageEventHandler
SpinnakerC.h, [354](#)
- spinCameraRegisterImageEventHandlerEx
SpinnakerC.h, [355](#)
- spinCameraRegisterImageListEventHandler
SpinnakerC.h, [355](#)
- spinCameraRelease
SpinnakerC.h, [356](#)
- spinCameraUnregisterDeviceEventHandler
SpinnakerC.h, [356](#)
- spinCameraUnregisterImageEventHandler
SpinnakerC.h, [357](#)
- spinCameraUnregisterImageListEventHandler
SpinnakerC.h, [357](#)
- spinCameraWritePort
SpinnakerC.h, [358](#)
- spinCategoryGetFeatureByIndex
SpinnakerGenApiC.h, [457](#)
- spinCategoryGetNumFeatures
SpinnakerGenApiC.h, [457](#)
- spinCategoryReleaseNode
SpinnakerGenApiC.h, [458](#)
- spinCommandExecute
SpinnakerGenApiC.h, [458](#)
- spinCommandIsDone
SpinnakerGenApiC.h, [459](#)
- spinDeviceArrivalEventHandler
SpinnakerDefsC.h, [442](#)
- spinDeviceArrivalEventHandlerCreate
SpinnakerC.h, [358](#)
- spinDeviceArrivalEventHandlerDestroy
SpinnakerC.h, [358](#)
- spinDeviceEventData
SpinnakerDefsC.h, [442](#)
- spinDeviceEventFunction
SpinnakerDefsC.h, [442](#)
- spinDeviceEventGetId
SpinnakerC.h, [359](#)
- spinDeviceEventGetName
SpinnakerC.h, [359](#)
- spinDeviceEventGetPayloadData
SpinnakerC.h, [360](#)
- spinDeviceEventGetPayloadDataSize
SpinnakerC.h, [360](#)
- spinDeviceEventHandler
SpinnakerDefsC.h, [442](#)
- spinDeviceEventHandlerCreate
SpinnakerC.h, [361](#)
- spinDeviceEventHandlerDestroy
SpinnakerC.h, [361](#)
- spinDeviceRemovalEventHandler
SpinnakerDefsC.h, [442](#)
- spinDeviceRemovalEventHandlerCreate
SpinnakerC.h, [362](#)
- spinDeviceRemovalEventHandlerDestroy
SpinnakerC.h, [362](#)
- spinEnumerationEntryGetEnumValue
SpinnakerGenApiC.h, [459](#)
- spinEnumerationEntryGetIntValue
SpinnakerGenApiC.h, [460](#)
- spinEnumerationEntryGetSymbolic
SpinnakerGenApiC.h, [460](#)
- spinEnumerationGetCurrentEntry
SpinnakerGenApiC.h, [461](#)
- spinEnumerationGetEntryByIndex
SpinnakerGenApiC.h, [461](#)
- spinEnumerationGetEntryByName
SpinnakerGenApiC.h, [462](#)
- spinEnumerationGetNumEntries
SpinnakerGenApiC.h, [462](#)
- spinEnumerationReleaseNode
SpinnakerGenApiC.h, [463](#)
- spinEnumerationSetEnumValue
SpinnakerGenApiC.h, [463](#)
- spinEnumerationSetIntValue
SpinnakerGenApiC.h, [464](#)
- spinErrorGetLast
SpinnakerC.h, [363](#)
- spinErrorGetLastBuildDate
SpinnakerC.h, [363](#)
- spinErrorGetLastBuildTime
SpinnakerC.h, [364](#)
- spinErrorGetLastFileName
SpinnakerC.h, [364](#)
- spinErrorGetLastFullMessage
SpinnakerC.h, [365](#)
- spinErrorGetLastFunctionName
SpinnakerC.h, [365](#)
- spinErrorGetLastLineNumber
SpinnakerC.h, [366](#)
- spinErrorGetLastMessage
SpinnakerC.h, [366](#)
- spinFloatGetMax
SpinnakerGenApiC.h, [464](#)
- spinFloatGetMin
SpinnakerGenApiC.h, [465](#)
- spinFloatGetRepresentation
SpinnakerGenApiC.h, [465](#)
- spinFloatGetUnit
SpinnakerGenApiC.h, [466](#)
- spinFloatGetValue
SpinnakerGenApiC.h, [466](#)
- spinFloatGetValueEx
SpinnakerGenApiC.h, [467](#)
- spinFloatSetValue
SpinnakerGenApiC.h, [467](#)
- spinFloatSetValueEx
SpinnakerGenApiC.h, [468](#)
- spinImage
SpinnakerDefsC.h, [442](#)
- spinImageCalculateStatistics
SpinnakerC.h, [367](#)
- spinImageCheckCRC
SpinnakerC.h, [367](#)

- spinImageChunkDataGetFloatValue
 - SpinnakerC.h, [368](#)
- spinImageChunkDataGetIntValue
 - SpinnakerC.h, [368](#)
- spinImageCreate
 - SpinnakerC.h, [368](#)
- spinImageCreateEmpty
 - SpinnakerC.h, [369](#)
- spinImageCreateEx
 - SpinnakerC.h, [369](#)
- spinImageCreateEx2
 - SpinnakerC.h, [370](#)
- spinImageDeepCopy
 - SpinnakerC.h, [370](#)
- spinImageDestroy
 - SpinnakerC.h, [371](#)
- spinImageEventFunction
 - SpinnakerDefsC.h, [443](#)
- spinImageEventHandler
 - SpinnakerDefsC.h, [443](#)
- spinImageEventHandlerCreate
 - SpinnakerC.h, [371](#)
- spinImageEventHandlerDestroy
 - SpinnakerC.h, [372](#)
- spinImageGetBitsPerPixel
 - SpinnakerC.h, [372](#)
- spinImageGetBufferSize
 - SpinnakerC.h, [373](#)
- spinImageGetChunkLayoutID
 - SpinnakerC.h, [373](#)
- spinImageGetColorProcessing
 - SpinnakerC.h, [374](#)
- spinImageGetData
 - SpinnakerC.h, [374](#)
- spinImageGetFrameID
 - SpinnakerC.h, [375](#)
- spinImageGetHeight
 - SpinnakerC.h, [375](#)
- spinImageGetID
 - SpinnakerC.h, [376](#)
- spinImageGetOffsetX
 - SpinnakerC.h, [376](#)
- spinImageGetOffsetY
 - SpinnakerC.h, [377](#)
- spinImageGetPaddingX
 - SpinnakerC.h, [377](#)
- spinImageGetPaddingY
 - SpinnakerC.h, [378](#)
- spinImageGetPayloadType
 - SpinnakerC.h, [378](#)
- spinImageGetPixelFormat
 - SpinnakerC.h, [379](#)
- spinImageGetPixelFormatName
 - SpinnakerC.h, [379](#)
- spinImageGetPrivateData
 - SpinnakerC.h, [380](#)
- spinImageGetSize
 - SpinnakerC.h, [380](#)
- spinImageGetStatus
 - SpinnakerC.h, [381](#)
- spinImageGetStatusDescription
 - SpinnakerC.h, [381](#)
- spinImageGetStride
 - SpinnakerC.h, [382](#)
- spinImageGetTimeStamp
 - SpinnakerC.h, [382](#)
- spinImageGetTLPayloadType
 - SpinnakerC.h, [383](#)
- spinImageGetTLPixelFormat
 - SpinnakerC.h, [383](#)
- spinImageGetTLPixelFormatNamespace
 - SpinnakerC.h, [384](#)
- spinImageGetValidPayloadSize
 - SpinnakerC.h, [384](#)
- spinImageGetWidth
 - SpinnakerC.h, [385](#)
- spinImageHasCRC
 - SpinnakerC.h, [385](#)
- spinImageIsIncomplete
 - SpinnakerC.h, [386](#)
- spinImageList
 - SpinnakerDefsC.h, [443](#)
- spinImageListAppend
 - SpinnakerC.h, [386](#)
- spinImageListClear
 - SpinnakerC.h, [387](#)
- spinImageListCreateEmpty
 - SpinnakerC.h, [387](#)
- spinImageListDestroy
 - SpinnakerC.h, [388](#)
- spinImageListEventFunction
 - SpinnakerDefsC.h, [443](#)
- spinImageListEventHandler
 - SpinnakerDefsC.h, [443](#)
- spinImageListEventHandlerCreate
 - SpinnakerC.h, [388](#)
- spinImageListEventHandlerDestroy
 - SpinnakerC.h, [389](#)
- spinImageListGet
 - SpinnakerC.h, [389](#)
- spinImageListGetByPixelFormat
 - SpinnakerC.h, [390](#)
- spinImageListGetSize
 - SpinnakerC.h, [390](#)
- spinImageListLoad
 - SpinnakerC.h, [391](#)
- spinImageListRelease
 - SpinnakerC.h, [391](#)
- spinImageListRemove
 - SpinnakerC.h, [391](#)
- spinImageListRemoveByPixelFormat
 - SpinnakerC.h, [392](#)
- spinImageListSave
 - SpinnakerC.h, [392](#)
- spinImageProcessor
 - SpinnakerDefsC.h, [443](#)

- spinImageProcessorApplyGamma
 - SpinnakerC.h, [393](#)
- spinImageProcessorConvert
 - SpinnakerC.h, [393](#)
- spinImageProcessorConvertImageList
 - SpinnakerC.h, [394](#)
- spinImageProcessorCreate
 - SpinnakerC.h, [395](#)
- spinImageProcessorDestroy
 - SpinnakerC.h, [395](#)
- spinImageProcessorGetColorProcessing
 - SpinnakerC.h, [396](#)
- spinImageProcessorGetNumDecompressionThreads
 - SpinnakerC.h, [396](#)
- spinImageProcessorSetColorProcessing
 - SpinnakerC.h, [396](#)
- spinImageProcessorSetNumDecompressionThreads
 - SpinnakerC.h, [397](#)
- spinImageRelease
 - SpinnakerC.h, [397](#)
- spinImageReset
 - SpinnakerC.h, [398](#)
- spinImageResetEx
 - SpinnakerC.h, [398](#)
- spinImageSave
 - SpinnakerC.h, [399](#)
- spinImageSaveBmp
 - SpinnakerC.h, [400](#)
- spinImageSaveFromExt
 - SpinnakerC.h, [400](#)
- spinImageSaveJpeg
 - SpinnakerC.h, [400](#)
- spinImageSaveJpg2
 - SpinnakerC.h, [401](#)
- spinImageSavePgm
 - SpinnakerC.h, [401](#)
- spinImageSavePng
 - SpinnakerC.h, [402](#)
- spinImageSavePpm
 - SpinnakerC.h, [402](#)
- spinImageSaveTiff
 - SpinnakerC.h, [403](#)
- spinImageStatistics
 - SpinnakerDefsC.h, [444](#)
- spinImageStatisticsCreate
 - SpinnakerC.h, [403](#)
- spinImageStatisticsDestroy
 - SpinnakerC.h, [404](#)
- spinImageStatisticsDisableAll
 - SpinnakerC.h, [404](#)
- spinImageStatisticsEnableAll
 - SpinnakerC.h, [405](#)
- spinImageStatisticsEnableGreyOnly
 - SpinnakerC.h, [405](#)
- spinImageStatisticsEnableHslOnly
 - SpinnakerC.h, [406](#)
- spinImageStatisticsEnableRgbOnly
 - SpinnakerC.h, [406](#)
- spinImageStatisticsGetAll
 - SpinnakerC.h, [406](#)
- spinImageStatisticsGetChannelStatus
 - SpinnakerC.h, [407](#)
- spinImageStatisticsGetHistogram
 - SpinnakerC.h, [408](#)
- spinImageStatisticsGetMean
 - SpinnakerC.h, [408](#)
- spinImageStatisticsGetNumPixelValues
 - SpinnakerC.h, [409](#)
- spinImageStatisticsGetPixelValueRange
 - SpinnakerC.h, [409](#)
- spinImageStatisticsGetRange
 - SpinnakerC.h, [410](#)
- spinImageStatisticsSetChannelStatus
 - SpinnakerC.h, [410](#)
- spinIntegerGetInc
 - SpinnakerGenApiC.h, [468](#)
- spinIntegerGetMax
 - SpinnakerGenApiC.h, [469](#)
- spinIntegerGetMin
 - SpinnakerGenApiC.h, [469](#)
- spinIntegerGetRepresentation
 - SpinnakerGenApiC.h, [470](#)
- spinIntegerGetValue
 - SpinnakerGenApiC.h, [470](#)
- spinIntegerGetValueEx
 - SpinnakerGenApiC.h, [471](#)
- spinIntegerSetValue
 - SpinnakerGenApiC.h, [471](#)
- spinIntegerSetValueEx
 - SpinnakerGenApiC.h, [472](#)
- spinInterface
 - SpinnakerDefsC.h, [444](#)
- spinInterfaceEventHandler
 - SpinnakerDefsC.h, [444](#)
- spinInterfaceEventHandlerCreate
 - SpinnakerC.h, [411](#)
- spinInterfaceEventHandlerDestroy
 - SpinnakerC.h, [411](#)
- spinInterfaceGetCameras
 - SpinnakerC.h, [412](#)
- spinInterfaceGetCamerasEx
 - SpinnakerC.h, [412](#)
- spinInterfaceGetTLNodeMap
 - SpinnakerC.h, [413](#)
- spinInterfaceIsInUse
 - SpinnakerC.h, [413](#)
- spinInterfaceList
 - SpinnakerDefsC.h, [444](#)
- spinInterfaceListClear
 - SpinnakerC.h, [414](#)
- spinInterfaceListCreateEmpty
 - SpinnakerC.h, [414](#)
- spinInterfaceListDestroy
 - SpinnakerC.h, [415](#)
- spinInterfaceListGet
 - SpinnakerC.h, [415](#)

- spinInterfaceListGetSize
 SpinnakerC.h, [416](#)
- spinInterfaceRegisterDeviceArrivalEventHandler
 SpinnakerC.h, [416](#)
- spinInterfaceRegisterDeviceRemovalEventHandler
 SpinnakerC.h, [417](#)
- spinInterfaceRegisterInterfaceEventHandler
 SpinnakerC.h, [417](#)
- spinInterfaceRelease
 SpinnakerC.h, [418](#)
- spinInterfaceSendActionCommand
 SpinnakerC.h, [418](#)
- spinInterfaceUnregisterDeviceArrivalEventHandler
 SpinnakerC.h, [419](#)
- spinInterfaceUnregisterDeviceRemovalEventHandler
 SpinnakerC.h, [419](#)
- spinInterfaceUnregisterInterfaceEventHandler
 SpinnakerC.h, [420](#)
- spinInterfaceUpdateCameras
 SpinnakerC.h, [420](#)
- spinLogDataGetCategoryName
 SpinnakerC.h, [421](#)
- spinLogDataGetLogMessage
 SpinnakerC.h, [421](#)
- spinLogDataGetNDC
 SpinnakerC.h, [422](#)
- spinLogDataGetPriority
 SpinnakerC.h, [422](#)
- spinLogDataGetPriorityName
 SpinnakerC.h, [423](#)
- spinLogDataGetThreadName
 SpinnakerC.h, [423](#)
- spinLogDataGetTimestamp
 SpinnakerC.h, [424](#)
- spinLogEventData
 SpinnakerDefsC.h, [444](#)
- spinLogEventFunction
 SpinnakerDefsC.h, [444](#)
- spinLogEventHandler
 SpinnakerDefsC.h, [445](#)
- spinLogEventHandlerCreate
 SpinnakerC.h, [424](#)
- spinLogEventHandlerDestroy
 SpinnakerC.h, [425](#)
- Spinnaker C API, [26](#)
- Spinnaker C Definitions, [21](#)
- Spinnaker C Enumerations, [45](#)
- Spinnaker C Function Signatures, [44](#)
- Spinnaker C GenICam API, [47](#)
- Spinnaker C GenICam Enumerations, [61](#)
- Spinnaker C GenICam Handles, [60](#)
- Spinnaker C Handles, [43](#)
- Spinnaker C QuickSpin API, [24](#)
- Spinnaker C Structures, [46](#)
- SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LASER
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_ACTION_COMMAND_STATUS_ERROR
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_ACTION_COMMAND_STATUS_OK
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILINEAR
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECTIONAL_FILTER
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSING
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQ_LINEAR
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPP
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR_FILTER
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NONE
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGOROUS
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEIGHTED_DIRECTIONAL_FILTER
 SpinnakerDefsC.h, [446](#)
- SPINNAKER_ERR_ABORT
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_ACCESS_DENIED
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_BUFFER_TOO_SMALL
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_BUSY
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_CUSTOM_ID
 SpinnakerDefsC.h, [448](#)
- SPINNAKER_ERR_ERROR
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_ACCESS
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_BAD_ALLOCATION
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_DYNAMIC_CAST
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_GENERIC
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_LOGICAL
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_OUT_OF_RANGE
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_PROPERTY
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_RUN_TIME
 SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_GENICAM_TIMEOUT
 SpinnakerDefsC.h, [447](#)

- SPINNAKER_ERR_IM_COLOR_CONVERSION
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_IM_CONVERT
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_IM_COPY
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_IM_HISTOGRAM_MEAN
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_IM_HISTOGRAM_RANGE
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_IM_MALLOC
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_IM_MIN_MAX
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_IM_NOT_SUPPORTED
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_INVALID_ADDRESS
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_INVALID_BUFFER
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_INVALID_HANDLE
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_INVALID_ID
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_INVALID_INDEX
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_INVALID_PARAMETER
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_INVALID_VALUE
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_IO
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_NO_DATA
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_NOT_AVAILABLE
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_NOT_IMPLEMENTED
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_NOT_INITIALIZED
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_OUT_OF_MEMORY
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_PARSING_CHUNK_DATA
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_RESOURCE_EXHAUSTED
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_RESOURCE_IN_USE
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_SUCCESS
SpinnakerDefsC.h, [447](#)
- SPINNAKER_ERR_TIMEOUT
SpinnakerDefsC.h, [447](#)
- SPINNAKER_IMAGE_FILE_FORMAT_BMP
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PGM
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PNG
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PPM
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_RAW
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_FILE_FORMAT_TIFF
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID
SpinnakerDefsC.h, [449](#)
- SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE
SpinnakerDefsC.h, [449](#)
- SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT
SpinnakerDefsC.h, [449](#)
- SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_STATUS_MISSING_LEADER
SpinnakerDefsC.h, [449](#)
- SPINNAKER_IMAGE_STATUS_MISSING_PACKETS
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_STATUS_MISSING_TRAILER
SpinnakerDefsC.h, [449](#)
- SPINNAKER_IMAGE_STATUS_NO_ERROR
SpinnakerDefsC.h, [448](#)
- SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES
SpinnakerDefsC.h, [449](#)
- SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT
SpinnakerDefsC.h, [449](#)
- SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT
SpinnakerDefsC.h, [449](#)
- SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR
SpinnakerDefsC.h, [448](#)
- SPINNAKER_LOG_LEVEL_ALERT
SpinnakerDefsC.h, [449](#)
- SPINNAKER_LOG_LEVEL_CRIT
SpinnakerDefsC.h, [449](#)
- SPINNAKER_LOG_LEVEL_DEBUG
SpinnakerDefsC.h, [449](#)
- SPINNAKER_LOG_LEVEL_ERROR
SpinnakerDefsC.h, [449](#)
- SPINNAKER_LOG_LEVEL_FATAL
SpinnakerDefsC.h, [449](#)
- SPINNAKER_LOG_LEVEL_INFO
SpinnakerDefsC.h, [449](#)
- SPINNAKER_LOG_LEVEL_NOTICE
SpinnakerDefsC.h, [449](#)
- SPINNAKER_LOG_LEVEL_NOTSET
SpinnakerDefsC.h, [449](#)

- SPINNAKER_LOG_LEVEL_OFF
SpinnakerDefsC.h, [449](#)
- SPINNAKER_LOG_LEVEL_WARN
SpinnakerDefsC.h, [449](#)
- SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID
SpinnakerDefsC.h, [451](#)
- SPINNAKER_STATISTICS_CHANNEL_BLUE
SpinnakerDefsC.h, [450](#)
- SPINNAKER_STATISTICS_CHANNEL_GREEN
SpinnakerDefsC.h, [450](#)
- SPINNAKER_STATISTICS_CHANNEL_GREY
SpinnakerDefsC.h, [450](#)
- SPINNAKER_STATISTICS_CHANNEL_HUE
SpinnakerDefsC.h, [450](#)
- SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS
SpinnakerDefsC.h, [450](#)
- SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNEL
SpinnakerDefsC.h, [450](#)
- SPINNAKER_STATISTICS_CHANNEL_RED
SpinnakerDefsC.h, [450](#)
- SPINNAKER_STATISTICS_CHANNEL_SATURATION
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_JPG
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_LZW
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_NONE
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID
SpinnakerDefsC.h, [451](#)
- SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_FILE
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_H264
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_IMAGE
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG2000
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED
SpinnakerDefsC.h, [451](#)
- SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED
SpinnakerDefsC.h, [451](#)
- SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED
SpinnakerDefsC.h, [451](#)
- SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN
SpinnakerDefsC.h, [450](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV
SpinnakerDefsC.h, [451](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC
SpinnakerDefsC.h, [451](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT
SpinnakerDefsC.h, [451](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT
SpinnakerDefsC.h, [451](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN
SpinnakerDefsC.h, [451](#)
- SpinnakerC.h
 - spinCameraBeginAcquisition, [340](#)
 - spinCameraDelInit, [341](#)
 - spinCameraDiscoverMaxPacketSize, [341](#)
 - spinCameraEndAcquisition, [342](#)
 - spinCameraForceIP, [342](#)
 - spinCameraGetAccessMode, [342](#)
 - spinCameraGetGuiXml, [343](#)
 - spinCameraGetNextImage, [343](#)
 - spinCameraGetNextImageEx, [344](#)
 - spinCameraGetNextImageSync, [344](#)
 - spinCameraGetNodeMap, [345](#)
 - spinCameraGetTLDeviceNodeMap, [346](#)
 - spinCameraGetTLStreamNodeMap, [346](#)
 - spinCameraGetUniqueID, [346](#)
 - spinCameraInit, [347](#)
 - spinCameraIsInitialized, [347](#)
 - spinCameraIsStreaming, [348](#)
 - spinCameraIsValid, [348](#)
 - spinCameraListAppend, [349](#)
 - spinCameraListClear, [349](#)
 - spinCameraListCreateEmpty, [350](#)
 - spinCameraListDestroy, [350](#)
 - spinCameraListGet, [351](#)
 - spinCameraListGetBySerial, [351](#)
 - spinCameraListGetSize, [352](#)
 - spinCameraListRemove, [352](#)
 - spinCameraListRemoveBySerial, [353](#)
 - spinCameraReadPort, [353](#)
 - spinCameraRegisterDeviceEventHandler, [353](#)
 - spinCameraRegisterDeviceEventHandlerEx, [354](#)
 - spinCameraRegisterImageEventHandler, [354](#)
 - spinCameraRegisterImageEventHandlerEx, [355](#)
 - spinCameraRegisterImageListEventHandler, [355](#)
 - spinCameraRelease, [356](#)
 - spinCameraUnregisterDeviceEventHandler, [356](#)
 - spinCameraUnregisterImageEventHandler, [357](#)
 - spinCameraUnregisterImageListEventHandler, [357](#)

- spinCameraWritePort, 358
- spinDeviceArrivalEventHandlerCreate, 358
- spinDeviceArrivalEventHandlerDestroy, 358
- spinDeviceEventGetId, 359
- spinDeviceEventGetName, 359
- spinDeviceEventGetPayloadData, 360
- spinDeviceEventGetPayloadDataSize, 360
- spinDeviceEventHandlerCreate, 361
- spinDeviceEventHandlerDestroy, 361
- spinDeviceRemovalEventHandlerCreate, 362
- spinDeviceRemovalEventHandlerDestroy, 362
- spinErrorGetLast, 363
- spinErrorGetLastBuildDate, 363
- spinErrorGetLastBuildTime, 364
- spinErrorGetLastFileName, 364
- spinErrorGetLastFullMessage, 365
- spinErrorGetLastFunctionName, 365
- spinErrorGetLastLineNumber, 366
- spinErrorGetLastMessage, 366
- spinImageCalculateStatistics, 367
- spinImageCheckCRC, 367
- spinImageChunkDataGetFloatValue, 368
- spinImageChunkDataGetIntValue, 368
- spinImageCreate, 368
- spinImageCreateEmpty, 369
- spinImageCreateEx, 369
- spinImageCreateEx2, 370
- spinImageDeepCopy, 370
- spinImageDestroy, 371
- spinImageEventHandlerCreate, 371
- spinImageEventHandlerDestroy, 372
- spinImageGetBitsPerPixel, 372
- spinImageGetBufferSize, 373
- spinImageGetChunkLayoutID, 373
- spinImageGetColorProcessing, 374
- spinImageGetData, 374
- spinImageGetFrameID, 375
- spinImageGetHeight, 375
- spinImageGetID, 376
- spinImageGetOffsetX, 376
- spinImageGetOffsetY, 377
- spinImageGetPaddingX, 377
- spinImageGetPaddingY, 378
- spinImageGetPayloadType, 378
- spinImageGetPixelFormat, 379
- spinImageGetPixelFormatName, 379
- spinImageGetPrivateData, 380
- spinImageGetSize, 380
- spinImageGetStatus, 381
- spinImageGetStatusDescription, 381
- spinImageGetStride, 382
- spinImageGetTimeStamp, 382
- spinImageGetTLPayloadType, 383
- spinImageGetTLPixelFormat, 383
- spinImageGetTLPixelFormatNamespace, 384
- spinImageGetValidPayloadSize, 384
- spinImageGetWidth, 385
- spinImageHasCRC, 385
- spinImageIsIncomplete, 386
- spinImageListAppend, 386
- spinImageListClear, 387
- spinImageListCreateEmpty, 387
- spinImageListDestroy, 388
- spinImageListEventHandlerCreate, 388
- spinImageListEventHandlerDestroy, 389
- spinImageListGet, 389
- spinImageListGetByPixelFormat, 390
- spinImageListGetSize, 390
- spinImageListLoad, 391
- spinImageListRelease, 391
- spinImageListRemove, 391
- spinImageListRemoveByPixelFormat, 392
- spinImageListSave, 392
- spinImageProcessorApplyGamma, 393
- spinImageProcessorConvert, 393
- spinImageProcessorConvertImageList, 394
- spinImageProcessorCreate, 395
- spinImageProcessorDestroy, 395
- spinImageProcessorGetColorProcessing, 396
- spinImageProcessorGetNumDecompression-Threads, 396
- spinImageProcessorSetColorProcessing, 396
- spinImageProcessorSetNumDecompression-Threads, 397
- spinImageRelease, 397
- spinImageReset, 398
- spinImageResetEx, 398
- spinImageSave, 399
- spinImageSaveBmp, 400
- spinImageSaveFromExt, 400
- spinImageSaveJpeg, 400
- spinImageSaveJpg2, 401
- spinImageSavePgm, 401
- spinImageSavePng, 402
- spinImageSavePpm, 402
- spinImageSaveTiff, 403
- spinImageStatisticsCreate, 403
- spinImageStatisticsDestroy, 404
- spinImageStatisticsDisableAll, 404
- spinImageStatisticsEnableAll, 405
- spinImageStatisticsEnableGreyOnly, 405
- spinImageStatisticsEnableHslOnly, 406
- spinImageStatisticsEnableRgbOnly, 406
- spinImageStatisticsGetAll, 406
- spinImageStatisticsGetChannelStatus, 407
- spinImageStatisticsGetHistogram, 408
- spinImageStatisticsGetMean, 408
- spinImageStatisticsGetNumPixelValues, 409
- spinImageStatisticsGetPixelValueRange, 409
- spinImageStatisticsGetRange, 410
- spinImageStatisticsSetChannelStatus, 410
- spinInterfaceEventHandlerCreate, 411
- spinInterfaceEventHandlerDestroy, 411
- spinInterfaceGetCameras, 412
- spinInterfaceGetCamerasEx, 412
- spinInterfaceGetTLNodeMap, 413

- spinInterfacelsInUse, 413
- spinInterfaceListClear, 414
- spinInterfaceListCreateEmpty, 414
- spinInterfaceListDestroy, 415
- spinInterfaceListGet, 415
- spinInterfaceListGetSize, 416
- spinInterfaceRegisterDeviceArrivalEventHandler, 416
- spinInterfaceRegisterDeviceRemovalEventHandler, 417
- spinInterfaceRegisterInterfaceEventHandler, 417
- spinInterfaceRelease, 418
- spinInterfaceSendActionCommand, 418
- spinInterfaceUnregisterDeviceArrivalEventHandler, 419
- spinInterfaceUnregisterDeviceRemovalEventHandler, 419
- spinInterfaceUnregisterInterfaceEventHandler, 420
- spinInterfaceUpdateCameras, 420
- spinLogDataGetCategoryName, 421
- spinLogDataGetLogMessage, 421
- spinLogDataGetNDC, 422
- spinLogDataGetPriority, 422
- spinLogDataGetPriorityName, 423
- spinLogDataGetThreadName, 423
- spinLogDataGetTimestamp, 424
- spinLogEventHandlerCreate, 424
- spinLogEventHandlerDestroy, 425
- spinSystemGetCameras, 425
- spinSystemGetCamerasEx, 426
- spinSystemGetInstance, 426
- spinSystemGetInterfaces, 427
- spinSystemGetLibraryVersion, 427
- spinSystemGetLoggingLevel, 427
- spinSystemGetTLNodeMap, 428
- spinSystemIsInUse, 428
- spinSystemRegisterDeviceArrivalEventHandler, 429
- spinSystemRegisterDeviceRemovalEventHandler, 429
- spinSystemRegisterInterfaceEventHandler, 430
- spinSystemRegisterLogEventHandler, 430
- spinSystemReleaseInstance, 431
- spinSystemSendActionCommand, 431
- spinSystemSetLoggingLevel, 432
- spinSystemUnregisterAllLogEventHandlers, 433
- spinSystemUnregisterDeviceArrivalEventHandler, 433
- spinSystemUnregisterDeviceRemovalEventHandler, 434
- spinSystemUnregisterInterfaceEventHandler, 434
- spinSystemUnregisterLogEventHandler, 435
- spinSystemUpdateCameras, 435
- spinSystemUpdateCamerasEx, 436
- SPINNAKER_API
 - SpinnakerPlatformC.h, 504
- SpinnakerDefsC.h
 - _spinActionCommandStatus, 445
 - _spinColorProcessingAlgorithm, 446
 - _spinError, 446
 - _spinImageFileFormat, 448
 - _spinImageStatus, 448
 - _spinLogLevel, 449
 - _spinStatisticsChannel, 449
 - _spinTIFFCompressionMethod, 450
 - _spinTLPayloadType, 450
 - _spinTLPixelFormatNamespace, 451
 - bool8_t, 441
 - False, 451
 - spinArrivalEventFunction, 441
 - spinCamera, 441
 - spinCameraList, 441
 - spinDeviceArrivalEventHandler, 442
 - spinDeviceEventData, 442
 - spinDeviceEventFunction, 442
 - spinDeviceEventHandler, 442
 - spinDeviceRemovalEventHandler, 442
 - spinImage, 442
 - spinImageEventFunction, 443
 - spinImageEventHandler, 443
 - spinImageList, 443
 - spinImageListEventFunction, 443
 - spinImageListEventHandler, 443
 - spinImageProcessor, 443
 - spinImageStatistics, 444
 - spinInterface, 444
 - spinInterfaceEventHandler, 444
 - spinInterfaceList, 444
 - spinLogEventData, 444
 - spinLogEventFunction, 444
 - spinLogEventHandler, 445
 - SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE, 446
 - SPINNAKER_ACTION_COMMAND_STATUS_ERROR, 446
 - SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME, 446
 - SPINNAKER_ACTION_COMMAND_STATUS_OK, 446
 - SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW, 446
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILINEAR, 446
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECTIONAL, 446
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSITIVE, 446
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQ_LINEAR, 446
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPP, 446
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR, 446
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR, 446
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_NONE, 446

- 446
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGOROUS, 446
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEIGHTED_DIRECTIONAL_FILTER, 446
- SPINNAKER_ERR_ABORT, 447
- SPINNAKER_ERR_ACCESS_DENIED, 447
- SPINNAKER_ERR_BUFFER_TOO_SMALL, 447
- SPINNAKER_ERR_BUSY, 447
- SPINNAKER_ERR_CUSTOM_ID, 448
- SPINNAKER_ERR_ERROR, 447
- SPINNAKER_ERR_GENICAM_ACCESS, 447
- SPINNAKER_ERR_GENICAM_BAD_ALLOCATION, 447
- SPINNAKER_ERR_GENICAM_DYNAMIC_CAST, 447
- SPINNAKER_ERR_GENICAM_GENERIC, 447
- SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT, 447
- SPINNAKER_ERR_GENICAM_LOGICAL, 447
- SPINNAKER_ERR_GENICAM_OUT_OF_RANGE, 447
- SPINNAKER_ERR_GENICAM_PROPERTY, 447
- SPINNAKER_ERR_GENICAM_RUN_TIME, 447
- SPINNAKER_ERR_GENICAM_TIMEOUT, 447
- SPINNAKER_ERR_IM_COLOR_CONVERSION, 447
- SPINNAKER_ERR_IM_CONVERT, 447
- SPINNAKER_ERR_IM_COPY, 447
- SPINNAKER_ERR_IM_HISTOGRAM_MEAN, 447
- SPINNAKER_ERR_IM_HISTOGRAM_RANGE, 447
- SPINNAKER_ERR_IM_MALLOC, 447
- SPINNAKER_ERR_IM_MIN_MAX, 447
- SPINNAKER_ERR_IM_NOT_SUPPORTED, 447
- SPINNAKER_ERR_INVALID_ADDRESS, 447
- SPINNAKER_ERR_INVALID_BUFFER, 447
- SPINNAKER_ERR_INVALID_HANDLE, 447
- SPINNAKER_ERR_INVALID_ID, 447
- SPINNAKER_ERR_INVALID_INDEX, 447
- SPINNAKER_ERR_INVALID_PARAMETER, 447
- SPINNAKER_ERR_INVALID_VALUE, 447
- SPINNAKER_ERR_IO, 447
- SPINNAKER_ERR_NO_DATA, 447
- SPINNAKER_ERR_NOT_AVAILABLE, 447
- SPINNAKER_ERR_NOT_IMPLEMENTED, 447
- SPINNAKER_ERR_NOT_INITIALIZED, 447
- SPINNAKER_ERR_OUT_OF_MEMORY, 447
- SPINNAKER_ERR_PARSING_CHUNK_DATA, 447
- SPINNAKER_ERR_RESOURCE_EXHAUSTED, 447
- SPINNAKER_ERR_RESOURCE_IN_USE, 447
- SPINNAKER_ERR_SUCCESS, 447
- SPINNAKER_ERR_TIMEOUT, 447
- SPINNAKER_IMAGE_FILE_FORMAT_BMP, 448
- SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS, 448
- SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT, 448
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG, 448
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000, 448
- SPINNAKER_IMAGE_FILE_FORMAT_PGM, 448
- SPINNAKER_IMAGE_FILE_FORMAT_PNG, 448
- SPINNAKER_IMAGE_FILE_FORMAT_PPM, 448
- SPINNAKER_IMAGE_FILE_FORMAT_RAW, 448
- SPINNAKER_IMAGE_FILE_FORMAT_TIFF, 448
- SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID, 449
- SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED, 448
- SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE, 449
- SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW, 448
- SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT, 449
- SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT, 448
- SPINNAKER_IMAGE_STATUS_MISSING_LEADER, 449
- SPINNAKER_IMAGE_STATUS_MISSING_PACKETS, 448
- SPINNAKER_IMAGE_STATUS_MISSING_TRAILER, 449
- SPINNAKER_IMAGE_STATUS_NO_ERROR, 448
- SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES, 449
- SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT, 449
- SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT, 449
- SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR, 448
- SPINNAKER_LOG_LEVEL_ALERT, 449
- SPINNAKER_LOG_LEVEL_CRIT, 449
- SPINNAKER_LOG_LEVEL_DEBUG, 449
- SPINNAKER_LOG_LEVEL_ERROR, 449
- SPINNAKER_LOG_LEVEL_FATAL, 449
- SPINNAKER_LOG_LEVEL_INFO, 449
- SPINNAKER_LOG_LEVEL_NOTICE, 449
- SPINNAKER_LOG_LEVEL_NOTSET, 449
- SPINNAKER_LOG_LEVEL_OFF, 449
- SPINNAKER_LOG_LEVEL_WARN, 449
- SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID, 451
- SPINNAKER_STATISTICS_CHANNEL_BLUE, 450
- SPINNAKER_STATISTICS_CHANNEL_GREEN, 450
- SPINNAKER_STATISTICS_CHANNEL_GREY, 450
- SPINNAKER_STATISTICS_CHANNEL_HUE, 450
- SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS, 450

- SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS, 450
- SPINNAKER_STATISTICS_CHANNEL_RED, 450
- SPINNAKER_STATISTICS_CHANNEL_SATURATIONS, 450
- SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE, 450
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3, 450
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4, 450
- SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE, 450
- SPINNAKER_TIFF_COMPRESS_METHOD_JPG, 450
- SPINNAKER_TIFF_COMPRESS_METHOD_LZW, 450
- SPINNAKER_TIFF_COMPRESS_METHOD_NONE, 450
- SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS, 450
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA, 450
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY, 450
- SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID, 451
- SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC, 450
- SPINNAKER_TLPAYLOAD_TYPE_FILE, 450
- SPINNAKER_TLPAYLOAD_TYPE_H264, 450
- SPINNAKER_TLPAYLOAD_TYPE_IMAGE, 450
- SPINNAKER_TLPAYLOAD_TYPE_JPEG, 450
- SPINNAKER_TLPAYLOAD_TYPE_JPEG2000, 450
- SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED, 451
- SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED, 451
- SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED, 451
- SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART, 450
- SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA, 450
- SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN, 450
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV, 451
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC, 451
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT, 451
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT, 451
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN, 451
- spinRemovalEventFunction, 445
- SpinSystem, 445
- spinVideo, 445
- True, 451
- SpinakerGenApiC.h
 - spinBooleanGetValue, 456
 - spinBooleanSetValue, 456
 - spinCategoryGetFeatureByIndex, 457
 - spinCategoryGetNumFeatures, 457
 - spinCategoryReleaseNode, 458
 - spinCommandExecute, 458
 - spinCommandIsDone, 459
 - spinEnumerationEntryGetEnumValue, 459
 - spinEnumerationEntryGetIntValue, 460
 - spinEnumerationEntryGetSymbolic, 460
 - spinEnumerationGetCurrentEntry, 461
 - spinEnumerationGetEntryByIndex, 461
 - spinEnumerationGetEntryByName, 462
 - spinEnumerationGetNumEntries, 462
 - spinEnumerationReleaseNode, 463
 - spinEnumerationSetEnumValue, 463
 - spinEnumerationSetIntValue, 464
 - spinFloatGetMax, 464
 - spinFloatGetMin, 465
 - spinFloatGetRepresentation, 465
 - spinFloatGetUnit, 466
 - spinFloatGetValue, 466
 - spinFloatGetValueEx, 467
 - spinFloatSetValue, 467
 - spinFloatSetValueEx, 468
 - spinIntegerGetInc, 468
 - spinIntegerGetMax, 469
 - spinIntegerGetMin, 469
 - spinIntegerGetRepresentation, 470
 - spinIntegerGetValue, 470
 - spinIntegerGetValueEx, 471
 - spinIntegerSetValue, 471
 - spinIntegerSetValueEx, 472
 - spinNodeDeregisterCallback, 472
 - spinNodeFromString, 473
 - spinNodeFromStringEx, 473
 - spinNodeGetAccessMode, 474
 - spinNodeGetCachingMode, 474
 - spinNodeGetDescription, 475
 - spinNodeGetDisplayName, 475
 - spinNodeGetImposedAccessMode, 476
 - spinNodeGetImposedVisibility, 476
 - spinNodeGetName, 477
 - spinNodeGetNameSpace, 477
 - spinNodeGetPollingTime, 478
 - spinNodeGetToolTip, 478
 - spinNodeGetType, 479
 - spinNodeGetVisibility, 479
 - spinNodeInvalidateNode, 480
 - spinNodesAvailable, 480
 - spinNodesEqual, 481
 - spinNodesImplemented, 481
 - spinNodesReadable, 482
 - spinNodesWritable, 482

- spinNodeMapGetNode, [483](#)
- spinNodeMapGetNodeByIndex, [483](#)
- spinNodeMapGetNumNodes, [484](#)
- spinNodeMapPoll, [484](#)
- spinNodeMapReleaseNode, [485](#)
- spinNodeRegisterCallback, [485](#)
- spinNodeToString, [486](#)
- spinNodeToStringEx, [486](#)
- spinRegisterGet, [487](#)
- spinRegisterGetAddress, [487](#)
- spinRegisterGetEx, [488](#)
- spinRegisterGetLength, [488](#)
- spinRegisterSet, [489](#)
- spinRegisterSetEx, [489](#)
- spinRegisterSetReference, [490](#)
- spinStringGetMaxLength, [490](#)
- spinStringGetValue, [491](#)
- spinStringGetValueEx, [491](#)
- spinStringSetValue, [492](#)
- spinStringSetValueEx, [492](#)
- SpinnakerGenApiDefsC.h
 - _CycleDetectAccesMode, [497](#)
 - _UndefinedAccesMode, [497](#)
 - _UndefinedCachingMode, [497](#)
 - _UndefinedEDisplayNotation, [497](#)
 - _UndefinedESlope, [502](#)
 - _UndefinedEXMLValidation, [503](#)
 - _UndefinedEndian, [498](#)
 - _UndefinedNameSpace, [500](#)
 - _UndefinedRepresentation, [501](#)
 - _UndefinedSign, [501](#)
 - _UndefinedStandardNameSpace, [502](#)
 - _UndefinedVisibility, [502](#)
 - _UndefinedYesNo, [503](#)
 - _spinAccessMode, [496](#)
 - _spinCachingMode, [497](#)
 - _spinDisplayNotation, [497](#)
 - _spinEndianess, [497](#)
 - _spinIncMode, [498](#)
 - _spinInputDirection, [498](#)
 - _spinInterfaceType, [498](#)
 - _spinLinkType, [499](#)
 - _spinNameSpace, [500](#)
 - _spinNodeType, [500](#)
 - _spinRepresentation, [501](#)
 - _spinSign, [501](#)
 - _spinSlope, [501](#)
 - _spinStandardNameSpace, [502](#)
 - _spinVisibility, [502](#)
 - _spinXMLValidation, [503](#)
 - _spinYesNo, [503](#)
- Automatic, [502](#)
- BaseNode, [500](#)
- Beginner, [502](#)
- BigEndian, [498](#)
- Boolean, [501](#)
- BooleanNode, [500](#)
- CategoryNode, [501](#)
- CL, [502](#)
- CommandNode, [500](#)
- ctAllDependingNodes, [500](#)
- ctAllTerminalNodes, [500](#)
- ctDependingChildren, [500](#)
- ctInvalidators, [500](#)
- ctReadingChildren, [500](#)
- ctWritingChildren, [500](#)
- Custom, [500](#)
- Decreasing, [502](#)
- EnumEntryNode, [501](#)
- EnumerationNode, [501](#)
- Expert, [502](#)
- fixedIncrement, [498](#)
- FloatNode, [500](#)
- fnAutomatic, [497](#)
- fnFixed, [497](#)
- fnScientific, [497](#)
- GEV, [502](#)
- Guru, [502](#)
- HexNumber, [501](#)
- idFrom, [498](#)
- idNone, [498](#)
- idTo, [498](#)
- IIDC, [502](#)
- Increasing, [502](#)
- IntegerNode, [500](#)
- intflBase, [499](#)
- intflBoolean, [499](#)
- intflCategory, [499](#)
- intflCommand, [499](#)
- intflEnumEntry, [499](#)
- intflEnumeration, [499](#)
- intflFloat, [499](#)
- intflInteger, [499](#)
- intflPort, [499](#)
- intflRegister, [499](#)
- intflString, [499](#)
- intflValue, [499](#)
- Invisible, [502](#)
- IPV4Address, [501](#)
- Linear, [501](#)
- listIncrement, [498](#)
- LittleEndian, [498](#)
- Logarithmic, [501](#)
- MACAddress, [501](#)
- NA, [497](#)
- NI, [497](#)
- No, [503](#)
- NoCache, [497](#)
- noIncrement, [498](#)
- None, [502](#)
- PortNode, [501](#)
- PureNumber, [501](#)
- RegisterNode, [501](#)
- RO, [497](#)
- RW, [497](#)
- Signed, [501](#)

- spinNodeCallbackFunction, [496](#)
- spinNodeCallbackHandle, [496](#)
- spinNodeHandle, [496](#)
- spinNodeMapHandle, [496](#)
- Standard, [500](#)
- StringNode, [500](#)
- UnknownNode, [501](#)
- Unsigned, [501](#)
- USB, [502](#)
- ValueNode, [500](#)
- Varying, [502](#)
- WO, [497](#)
- WriteAround, [497](#)
- WriteThrough, [497](#)
- xvAll, [503](#)
- xvCycles, [503](#)
- xvDefault, [503](#)
- xvLoad, [503](#)
- xvSFNC, [503](#)
- Yes, [503](#)
- SpinnakerPlatformC.h
 - SPINNAKERC_API, [504](#)
- spinNodeCallbackFunction
 - SpinnakerGenApiDefsC.h, [496](#)
- spinNodeCallbackHandle
 - SpinnakerGenApiDefsC.h, [496](#)
- spinNodeDeregisterCallback
 - SpinnakerGenApiC.h, [472](#)
- spinNodeFromString
 - SpinnakerGenApiC.h, [473](#)
- spinNodeFromStringEx
 - SpinnakerGenApiC.h, [473](#)
- spinNodeGetAccessMode
 - SpinnakerGenApiC.h, [474](#)
- spinNodeGetCachingMode
 - SpinnakerGenApiC.h, [474](#)
- spinNodeGetDescription
 - SpinnakerGenApiC.h, [475](#)
- spinNodeGetDisplayName
 - SpinnakerGenApiC.h, [475](#)
- spinNodeGetImposedAccessMode
 - SpinnakerGenApiC.h, [476](#)
- spinNodeGetImposedVisibility
 - SpinnakerGenApiC.h, [476](#)
- spinNodeGetName
 - SpinnakerGenApiC.h, [477](#)
- spinNodeGetNameSpace
 - SpinnakerGenApiC.h, [477](#)
- spinNodeGetPollingTime
 - SpinnakerGenApiC.h, [478](#)
- spinNodeGetToolTip
 - SpinnakerGenApiC.h, [478](#)
- spinNodeGetType
 - SpinnakerGenApiC.h, [479](#)
- spinNodeGetVisibility
 - SpinnakerGenApiC.h, [479](#)
- spinNodeHandle
 - SpinnakerGenApiDefsC.h, [496](#)
- spinNodeInvalidateNode
 - SpinnakerGenApiC.h, [480](#)
- spinNodesAvailable
 - SpinnakerGenApiC.h, [480](#)
- spinNodesEqual
 - SpinnakerGenApiC.h, [481](#)
- spinNodesImplemented
 - SpinnakerGenApiC.h, [481](#)
- spinNodesReadable
 - SpinnakerGenApiC.h, [482](#)
- spinNodesWritable
 - SpinnakerGenApiC.h, [482](#)
- spinNodeMapGetNode
 - SpinnakerGenApiC.h, [483](#)
- spinNodeMapGetNodeByIndex
 - SpinnakerGenApiC.h, [483](#)
- spinNodeMapGetNumNodes
 - SpinnakerGenApiC.h, [484](#)
- spinNodeMapHandle
 - SpinnakerGenApiDefsC.h, [496](#)
- spinNodeMapPoll
 - SpinnakerGenApiC.h, [484](#)
- spinNodeMapReleaseNode
 - SpinnakerGenApiC.h, [485](#)
- spinNodeRegisterCallback
 - SpinnakerGenApiC.h, [485](#)
- spinNodeToString
 - SpinnakerGenApiC.h, [486](#)
- spinNodeToStringEx
 - SpinnakerGenApiC.h, [486](#)
- spinRegisterGet
 - SpinnakerGenApiC.h, [487](#)
- spinRegisterGetAddress
 - SpinnakerGenApiC.h, [487](#)
- spinRegisterGetEx
 - SpinnakerGenApiC.h, [488](#)
- spinRegisterGetLength
 - SpinnakerGenApiC.h, [488](#)
- spinRegisterSet
 - SpinnakerGenApiC.h, [489](#)
- spinRegisterSetEx
 - SpinnakerGenApiC.h, [489](#)
- spinRegisterSetReference
 - SpinnakerGenApiC.h, [490](#)
- spinRemovalEventFunction
 - SpinnakerDefsC.h, [445](#)
- spinStringGetMaxLength
 - SpinnakerGenApiC.h, [490](#)
- spinStringGetValue
 - SpinnakerGenApiC.h, [491](#)
- spinStringGetValueEx
 - SpinnakerGenApiC.h, [491](#)
- spinStringSetValue
 - SpinnakerGenApiC.h, [492](#)
- spinStringSetValueEx
 - SpinnakerGenApiC.h, [492](#)
- spinSystem
 - SpinnakerDefsC.h, [445](#)

- spinSystemGetCameras
 - SpinnakerC.h, [425](#)
- spinSystemGetCamerasEx
 - SpinnakerC.h, [426](#)
- spinSystemGetInstance
 - SpinnakerC.h, [426](#)
- spinSystemGetInterfaces
 - SpinnakerC.h, [427](#)
- spinSystemGetLibraryVersion
 - SpinnakerC.h, [427](#)
- spinSystemGetLoggingLevel
 - SpinnakerC.h, [427](#)
- spinSystemGetTLNodeMap
 - SpinnakerC.h, [428](#)
- spinSystemIsInUse
 - SpinnakerC.h, [428](#)
- spinSystemRegisterDeviceArrivalEventHandler
 - SpinnakerC.h, [429](#)
- spinSystemRegisterDeviceRemovalEventHandler
 - SpinnakerC.h, [429](#)
- spinSystemRegisterInterfaceEventHandler
 - SpinnakerC.h, [430](#)
- spinSystemRegisterLogEventHandler
 - SpinnakerC.h, [430](#)
- spinSystemReleaseInstance
 - SpinnakerC.h, [431](#)
- spinSystemSendActionCommand
 - SpinnakerC.h, [431](#)
- spinSystemSetLoggingLevel
 - SpinnakerC.h, [432](#)
- spinSystemUnregisterAllLogEventHandlers
 - SpinnakerC.h, [433](#)
- spinSystemUnregisterDeviceArrivalEventHandler
 - SpinnakerC.h, [433](#)
- spinSystemUnregisterDeviceRemovalEventHandler
 - SpinnakerC.h, [434](#)
- spinSystemUnregisterInterfaceEventHandler
 - SpinnakerC.h, [434](#)
- spinSystemUnregisterLogEventHandler
 - SpinnakerC.h, [435](#)
- spinSystemUpdateCameras
 - SpinnakerC.h, [435](#)
- spinSystemUpdateCamerasEx
 - SpinnakerC.h, [436](#)
- spinVideo
 - SpinnakerDefsC.h, [445](#)
- SpinVideo Recording Access, [62](#)
- spinVideoAppend
 - SpinVideoC.h, [505](#)
- SpinVideoC.h
 - spinVideoAppend, [505](#)
 - spinVideoClose, [505](#)
 - spinVideoOpenH264, [506](#)
 - spinVideoOpenMJPEG, [506](#)
 - spinVideoOpenUncompressed, [506](#)
 - spinVideoSetMaximumFileSize, [506](#)
- spinVideoClose
 - SpinVideoC.h, [505](#)
- spinVideoOpenH264
 - SpinVideoC.h, [506](#)
- spinVideoOpenMJPEG
 - SpinVideoC.h, [506](#)
- spinVideoOpenUncompressed
 - SpinVideoC.h, [506](#)
- spinVideoSetMaximumFileSize
 - SpinVideoC.h, [506](#)
- Standard
 - SpinnakerGenApiDefsC.h, [500](#)
- Status
 - _actionCommandResult, [69](#)
- StreamAnnounceBufferMinimum
 - _quickSpinTLStream, [181](#)
- StreamAnnouncedBufferCount
 - _quickSpinTLStream, [181](#)
- StreamBlockTransferSize
 - _quickSpinTLStream, [181](#)
- StreamBufferAlignment
 - _quickSpinTLStream, [182](#)
- StreamBufferCountManual
 - _quickSpinTLStream, [182](#)
- StreamBufferCountMax
 - _quickSpinTLStream, [182](#)
- StreamBufferCountMode
 - _quickSpinTLStream, [182](#)
- StreamBufferCountMode_Manual
 - TransportLayerDefsC.h, [513](#)
- StreamBufferCountResult
 - _quickSpinTLStream, [182](#)
- StreamBufferHandlingMode
 - _quickSpinTLStream, [182](#)
- StreamBufferHandlingMode_NewestFirst
 - TransportLayerDefsC.h, [514](#)
- StreamBufferHandlingMode_NewestOnly
 - TransportLayerDefsC.h, [514](#)
- StreamBufferHandlingMode_OldestFirst
 - TransportLayerDefsC.h, [514](#)
- StreamBufferHandlingMode_OldestFirstOverwrite
 - TransportLayerDefsC.h, [514](#)
- StreamChunkCountMaximum
 - _quickSpinTLStream, [182](#)
- StreamCRCCheckEnable
 - _quickSpinTLStream, [182](#)
- StreamDeliveredFrameCount
 - _quickSpinTLStream, [183](#)
- StreamDroppedFrameCount
 - _quickSpinTLStream, [183](#)
- StreamID
 - _quickSpinTLStream, [183](#)
- StreamIncompleteFrameCount
 - _quickSpinTLStream, [183](#)
- StreamInputBufferCount
 - _quickSpinTLStream, [183](#)
- StreamIsGrabbing
 - _quickSpinTLStream, [183](#)
- StreamLostFrameCount
 - _quickSpinTLStream, [183](#)

- StreamMissedPacketCount
 - _quickSpinTLStream, 183
- StreamMode
 - _quickSpinTLStream, 184
- StreamMode_LWF
 - TransportLayerDefsC.h, 514
- StreamMode_MVA
 - TransportLayerDefsC.h, 514
- StreamMode_Socket
 - TransportLayerDefsC.h, 514
- StreamOutputBufferCount
 - _quickSpinTLStream, 184
- StreamPacketResendEnable
 - _quickSpinTLStream, 184
- StreamPacketResendMaxRequests
 - _quickSpinTLStream, 184
- StreamPacketResendReceivedPacketCount
 - _quickSpinTLStream, 184
- StreamPacketResendRequestCount
 - _quickSpinTLStream, 184
- StreamPacketResendRequestedPacketCount
 - _quickSpinTLStream, 184
- StreamPacketResendRequestSuccessCount
 - _quickSpinTLStream, 184
- StreamPacketResendTimeout
 - _quickSpinTLStream, 185
- StreamReceivedFrameCount
 - _quickSpinTLStream, 185
- StreamReceivedPacketCount
 - _quickSpinTLStream, 185
- StreamStartedFrameCount
 - _quickSpinTLStream, 185
- StreamType
 - _quickSpinTLStream, 185
- StreamType_CameraLink
 - TransportLayerDefsC.h, 515
- StreamType_CameraLinkHS
 - TransportLayerDefsC.h, 515
- StreamType_CoaXPress
 - TransportLayerDefsC.h, 515
- StreamType_Custom
 - TransportLayerDefsC.h, 515
- StreamType_GigEVision
 - TransportLayerDefsC.h, 515
- StreamType_USB3Vision
 - TransportLayerDefsC.h, 515
- String Access, 51
- StringNode
 - SpinnakerGenApiDefsC.h, 500
- System Access, 28
- Test0001
 - _quickSpin, 158
- TestEventGenerate
 - _quickSpin, 158
- TestPattern
 - _quickSpin, 158
- TestPattern_Increment
 - CameraDefsC.h, 315
- TestPattern_Off
 - CameraDefsC.h, 315
- TestPattern_SensorTestPattern
 - CameraDefsC.h, 315
- TestPatternGeneratorSelector
 - _quickSpin, 159
- TestPatternGeneratorSelector_PipelineStart
 - CameraDefsC.h, 315
- TestPatternGeneratorSelector_Sensor
 - CameraDefsC.h, 315
- TestPendingAck
 - _quickSpin, 159
- TimerDelay
 - _quickSpin, 159
- TimerDuration
 - _quickSpin, 159
- TimerReset
 - _quickSpin, 159
- TimerSelector
 - _quickSpin, 159
- TimerSelector_Timer0
 - CameraDefsC.h, 316
- TimerSelector_Timer1
 - CameraDefsC.h, 316
- TimerSelector_Timer2
 - CameraDefsC.h, 316
- TimerStatus
 - _quickSpin, 159
- TimerStatus_TimerActive
 - CameraDefsC.h, 316
- TimerStatus_TimerCompleted
 - CameraDefsC.h, 316
- TimerStatus_TimerIdle
 - CameraDefsC.h, 316
- TimerStatus_TimerTriggerWait
 - CameraDefsC.h, 316
- TimerTriggerActivation
 - _quickSpin, 159
- TimerTriggerActivation_AnyEdge
 - CameraDefsC.h, 316
- TimerTriggerActivation_FallingEdge
 - CameraDefsC.h, 316
- TimerTriggerActivation_LevelHigh
 - CameraDefsC.h, 316
- TimerTriggerActivation_LevelLow
 - CameraDefsC.h, 316
- TimerTriggerActivation_RisingEdge
 - CameraDefsC.h, 316
- TimerTriggerSource
 - _quickSpin, 160
- TimerTriggerSource_AcquisitionEnd
 - CameraDefsC.h, 317
- TimerTriggerSource_AcquisitionStart
 - CameraDefsC.h, 317
- TimerTriggerSource_AcquisitionTrigger
 - CameraDefsC.h, 317
- TimerTriggerSource_Action0
 - CameraDefsC.h, 318

- TimerTriggerSource_Action1
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_Action2
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_Counter0End
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Counter0Start
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Counter1End
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Counter1Start
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Counter2End
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Counter2Start
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Encoder0
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_Encoder1
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_Encoder2
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_ExposureEnd
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_ExposureStart
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_FrameBurstEnd
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_FrameBurstStart
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_FrameEnd
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_FrameStart
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_FrameTrigger
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Line0
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Line1
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Line2
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_LineEnd
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_LineStart
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_LineTrigger
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_LinkTrigger0
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_LinkTrigger1
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_LinkTrigger2
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_Off
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_SoftwareSignal0
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_SoftwareSignal1
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_SoftwareSignal2
 - CameraDefsC.h, [318](#)
- TimerTriggerSource_Timer0End
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Timer0Start
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Timer1End
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Timer1Start
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Timer2End
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_Timer2Start
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_UserOutput0
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_UserOutput1
 - CameraDefsC.h, [317](#)
- TimerTriggerSource_UserOutput2
 - CameraDefsC.h, [317](#)
- TimerValue
 - _quickSpin, [160](#)
- Timestamp
 - _quickSpin, [160](#)
- TimestampLatch
 - _quickSpin, [160](#)
- TimestampLatchValue
 - _quickSpin, [160](#)
- TimestampReset
 - _quickSpin, [160](#)
- TLDevice Structures, [64](#)
- TLDisplayName
 - _quickSpinTLSystem, [188](#)
- TLFileName
 - _quickSpinTLSystem, [189](#)
- TLID
 - _quickSpinTLSystem, [189](#)
- TLInterface Structures, [65](#)
- TLModelName
 - _quickSpinTLSystem, [189](#)
- TLParamsLocked
 - _quickSpin, [160](#)
- TLPath
 - _quickSpinTLSystem, [189](#)
- TLStream Structures, [66](#)
- TLSystem Structures, [67](#)
- TLType
 - _quickSpinTLSystem, [189](#)
- TLType_CameraLink
 - TransportLayerDefsC.h, [515](#)
- TLType_CameraLinkHS
 - TransportLayerDefsC.h, [515](#)
- TLType_CoaxPress
 - TransportLayerDefsC.h, [515](#)
- TLType_Custom
 - TransportLayerDefsC.h, [515](#)

- TLType_GigEVision
 - TransportLayerDefsC.h, [515](#)
- TLType_Mixed
 - TransportLayerDefsC.h, [515](#)
- TLType_USB3Vision
 - TransportLayerDefsC.h, [515](#)
- TLVendorName
 - _quickSpinTLSystem, [189](#)
- TLVersion
 - _quickSpinTLSystem, [189](#)
- TransferAbort
 - _quickSpin, [160](#)
- TransferBlockCount
 - _quickSpin, [161](#)
- TransferBurstCount
 - _quickSpin, [161](#)
- TransferComponentSelector
 - _quickSpin, [161](#)
- TransferComponentSelector_All
 - CameraDefsC.h, [318](#)
- TransferComponentSelector_Blue
 - CameraDefsC.h, [318](#)
- TransferComponentSelector_Green
 - CameraDefsC.h, [318](#)
- TransferComponentSelector_Red
 - CameraDefsC.h, [318](#)
- TransferControlMode
 - _quickSpin, [161](#)
- TransferControlMode_Automatic
 - CameraDefsC.h, [319](#)
- TransferControlMode_Basic
 - CameraDefsC.h, [319](#)
- TransferControlMode_UserControlled
 - CameraDefsC.h, [319](#)
- TransferOperationMode
 - _quickSpin, [161](#)
- TransferOperationMode_Continuous
 - CameraDefsC.h, [319](#)
- TransferOperationMode_MultiBlock
 - CameraDefsC.h, [319](#)
- TransferPause
 - _quickSpin, [161](#)
- TransferQueueCurrentBlockCount
 - _quickSpin, [161](#)
- TransferQueueMaxBlockCount
 - _quickSpin, [161](#)
- TransferQueueMode
 - _quickSpin, [162](#)
- TransferQueueMode_FirstInFirstOut
 - CameraDefsC.h, [319](#)
- TransferQueueOverflowCount
 - _quickSpin, [162](#)
- TransferResume
 - _quickSpin, [162](#)
- TransferSelector
 - _quickSpin, [162](#)
- TransferSelector_All
 - CameraDefsC.h, [319](#)
- TransferSelector_Stream0
 - CameraDefsC.h, [319](#)
- TransferSelector_Stream1
 - CameraDefsC.h, [319](#)
- TransferSelector_Stream2
 - CameraDefsC.h, [319](#)
- TransferStart
 - _quickSpin, [162](#)
- TransferStatus
 - _quickSpin, [162](#)
- TransferStatusSelector
 - _quickSpin, [162](#)
- TransferStatusSelector_Paused
 - CameraDefsC.h, [320](#)
- TransferStatusSelector_QueueOverflow
 - CameraDefsC.h, [320](#)
- TransferStatusSelector_Stopped
 - CameraDefsC.h, [320](#)
- TransferStatusSelector_Stopping
 - CameraDefsC.h, [320](#)
- TransferStatusSelector_Streaming
 - CameraDefsC.h, [320](#)
- TransferStop
 - _quickSpin, [162](#)
- TransferStreamChannel
 - _quickSpin, [163](#)
- TransferTriggerActivation
 - _quickSpin, [163](#)
- TransferTriggerActivation_AnyEdge
 - CameraDefsC.h, [320](#)
- TransferTriggerActivation_FallingEdge
 - CameraDefsC.h, [320](#)
- TransferTriggerActivation_LevelHigh
 - CameraDefsC.h, [320](#)
- TransferTriggerActivation_LevelLow
 - CameraDefsC.h, [320](#)
- TransferTriggerActivation_RisingEdge
 - CameraDefsC.h, [320](#)
- TransferTriggerMode
 - _quickSpin, [163](#)
- TransferTriggerMode_Off
 - CameraDefsC.h, [321](#)
- TransferTriggerMode_On
 - CameraDefsC.h, [321](#)
- TransferTriggerSelector
 - _quickSpin, [163](#)
- TransferTriggerSelector_TransferAbort
 - CameraDefsC.h, [321](#)
- TransferTriggerSelector_TransferActive
 - CameraDefsC.h, [321](#)
- TransferTriggerSelector_TransferBurstStart
 - CameraDefsC.h, [321](#)
- TransferTriggerSelector_TransferBurstStop
 - CameraDefsC.h, [321](#)
- TransferTriggerSelector_TransferPause
 - CameraDefsC.h, [321](#)
- TransferTriggerSelector_TransferResume
 - CameraDefsC.h, [321](#)

- TransferTriggerSelector_TransferStart
 - CameraDefsC.h, [321](#)
- TransferTriggerSelector_TransferStop
 - CameraDefsC.h, [321](#)
- TransferTriggerSource
 - _quickSpin, [163](#)
- TransferTriggerSource_Action0
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Action1
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Action2
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Counter0End
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Counter0Start
 - CameraDefsC.h, [321](#)
- TransferTriggerSource_Counter1End
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Counter1Start
 - CameraDefsC.h, [321](#)
- TransferTriggerSource_Counter2End
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Counter2Start
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Line0
 - CameraDefsC.h, [321](#)
- TransferTriggerSource_Line1
 - CameraDefsC.h, [321](#)
- TransferTriggerSource_Line2
 - CameraDefsC.h, [321](#)
- TransferTriggerSource_SoftwareSignal0
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_SoftwareSignal1
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_SoftwareSignal2
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Timer0End
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Timer0Start
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Timer1End
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Timer1Start
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Timer2End
 - CameraDefsC.h, [322](#)
- TransferTriggerSource_Timer2Start
 - CameraDefsC.h, [322](#)
- Transport Layer Enumerations, [63](#)
- TransportLayerDefsC.h
 - _spinTLDeviceAccessStatusEnums, [509](#)
 - _spinTLDeviceCurrentSpeedEnums, [509](#)
 - _spinTLDeviceEndiannessMechanismEnums, [509](#)
 - _spinTLDeviceTypeEnum, [511](#)
 - _spinTLFilterDriverStatusEnums, [511](#)
 - _spinTLGUIXMLLocationEnums, [512](#)
 - _spinTLGenICamXMLLocationEnums, [511](#)
 - _spinTLGevCCPEnums, [512](#)
 - _spinTLInterfaceTypeEnum, [512](#)
 - _spinTLPOEStatusEnums, [513](#)
 - _spinTLStreamBufferCountModeEnums, [513](#)
 - _spinTLStreamBufferHandlingModeEnums, [513](#)
 - _spinTLStreamModeEnums, [514](#)
 - _spinTLStreamTypeEnum, [514](#)
 - _spinTLTLTypeEnum, [515](#)
 - DeviceAccessStatus_Busy, [509](#)
 - DeviceAccessStatus_NoAccess, [509](#)
 - DeviceAccessStatus_OpenReadOnly, [509](#)
 - DeviceAccessStatus_OpenReadWrite, [509](#)
 - DeviceAccessStatus_ReadOnly, [509](#)
 - DeviceAccessStatus_ReadWrite, [509](#)
 - DeviceAccessStatus_Unknown, [509](#)
 - DeviceCurrentSpeed_FullSpeed, [509](#)
 - DeviceCurrentSpeed_HighSpeed, [509](#)
 - DeviceCurrentSpeed_LowSpeed, [509](#)
 - DeviceCurrentSpeed_SuperSpeed, [509](#)
 - DeviceCurrentSpeed_UnknownSpeed, [509](#)
 - DeviceEndiannessMechanism_Legacy, [511](#)
 - DeviceEndiannessMechanism_Standard, [511](#)
 - DeviceType_CameraLink, [511](#)
 - DeviceType_CameraLinkHS, [511](#)
 - DeviceType_CoaXPress, [511](#)
 - DeviceType_Custom, [511](#)
 - DeviceType_GigEVision, [511](#)
 - DeviceType_USB3Vision, [511](#)
 - FilterDriverStatus_Disabled, [511](#)
 - FilterDriverStatus_Enabled, [511](#)
 - FilterDriverStatus_NotSupported, [511](#)
 - GenICamXMLLocation_Device, [512](#)
 - GenICamXMLLocation_Host, [512](#)
 - GevCCP_EnumEntry_GevCCP_ControlAccess, [512](#)
 - GevCCP_EnumEntry_GevCCP_ExclusiveAccess, [512](#)
 - GevCCP_EnumEntry_GevCCP_OpenAccess, [512](#)
 - GUIXMLLocation_Device, [512](#)
 - GUIXMLLocation_Host, [512](#)
 - InterfaceType_CameraLink, [512](#)
 - InterfaceType_CameraLinkHS, [513](#)
 - InterfaceType_CoaXPress, [513](#)
 - InterfaceType_Custom, [513](#)
 - InterfaceType_GigEVision, [512](#)
 - InterfaceType_USB3Vision, [513](#)
 - NUMDEVICEACCESSSTATUS, [509](#)
 - NUMDEVICECURRENTSPEED, [509](#)
 - NUMDEVICEENDIANESSMECHANISM, [511](#)
 - NUMDEVICETYPE, [511](#)
 - NUMFILTERDRIVERSTATUS, [511](#)
 - NUMGENICAMXMLLOCATION, [512](#)
 - NUMGEVCCP, [512](#)
 - NUMGUIXMLLOCATION, [512](#)
 - NUMINTERFACETYPE, [513](#)
 - NUMPOESTATUS, [513](#)
 - NUMSTREAMBUFFERCOUNTMODE, [513](#)
 - NUMSTREAMBUFFERHANDLINGMODE, [514](#)
 - NUMSTREAMMODE, [514](#)

- NUMSTREAMTYPE, [515](#)
- NUMTLTYPE, [515](#)
- POEStatus_NotSupported, [513](#)
- POEStatus_PowerOff, [513](#)
- POEStatus_PowerOn, [513](#)
- StreamBufferCountMode_Manual, [513](#)
- StreamBufferHandlingMode_NewestFirst, [514](#)
- StreamBufferHandlingMode_NewestOnly, [514](#)
- StreamBufferHandlingMode_OldestFirst, [514](#)
- StreamBufferHandlingMode_OldestFirstOverwrite, [514](#)
- StreamMode_LWF, [514](#)
- StreamMode_MVA, [514](#)
- StreamMode_Socket, [514](#)
- StreamType_CameraLink, [515](#)
- StreamType_CameraLinkHS, [515](#)
- StreamType_CoaXPRESS, [515](#)
- StreamType_Custom, [515](#)
- StreamType_GigEVision, [515](#)
- StreamType_USB3Vision, [515](#)
- TLType_CameraLink, [515](#)
- TLType_CameraLinkHS, [515](#)
- TLType_CoaXPRESS, [515](#)
- TLType_Custom, [515](#)
- TLType_GigEVision, [515](#)
- TLType_Mixed, [515](#)
- TLType_USB3Vision, [515](#)
- TriggerActivation
 - _quickSpin, [163](#)
- TriggerActivation_AnyEdge
 - CameraDefsC.h, [322](#)
- TriggerActivation_FallingEdge
 - CameraDefsC.h, [322](#)
- TriggerActivation_LevelHigh
 - CameraDefsC.h, [322](#)
- TriggerActivation_LevelLow
 - CameraDefsC.h, [322](#)
- TriggerActivation_RisingEdge
 - CameraDefsC.h, [322](#)
- TriggerDelay
 - _quickSpin, [163](#)
- TriggerDivider
 - _quickSpin, [163](#)
- TriggerEventTest
 - _quickSpin, [164](#)
- TriggerMode
 - _quickSpin, [164](#)
- TriggerMode_Off
 - CameraDefsC.h, [323](#)
- TriggerMode_On
 - CameraDefsC.h, [323](#)
- TriggerMultiplier
 - _quickSpin, [164](#)
- TriggerOverlap
 - _quickSpin, [164](#)
- TriggerOverlap_Off
 - CameraDefsC.h, [323](#)
- TriggerOverlap_PreviousFrame
 - CameraDefsC.h, [323](#)
- TriggerOverlap_ReadOut
 - CameraDefsC.h, [323](#)
- TriggerSelector
 - _quickSpin, [164](#)
- TriggerSelector_AcquisitionStart
 - CameraDefsC.h, [323](#)
- TriggerSelector_FrameBurstStart
 - CameraDefsC.h, [323](#)
- TriggerSelector_FrameStart
 - CameraDefsC.h, [323](#)
- TriggerSoftware
 - _quickSpin, [164](#)
- TriggerSource
 - _quickSpin, [164](#)
- TriggerSource_Action0
 - CameraDefsC.h, [324](#)
- TriggerSource_Counter0End
 - CameraDefsC.h, [324](#)
- TriggerSource_Counter0Start
 - CameraDefsC.h, [324](#)
- TriggerSource_Counter1End
 - CameraDefsC.h, [324](#)
- TriggerSource_Counter1Start
 - CameraDefsC.h, [324](#)
- TriggerSource_Line0
 - CameraDefsC.h, [324](#)
- TriggerSource_Line1
 - CameraDefsC.h, [324](#)
- TriggerSource_Line2
 - CameraDefsC.h, [324](#)
- TriggerSource_Line3
 - CameraDefsC.h, [324](#)
- TriggerSource_LogicBlock0
 - CameraDefsC.h, [324](#)
- TriggerSource_LogicBlock1
 - CameraDefsC.h, [324](#)
- TriggerSource_Software
 - CameraDefsC.h, [324](#)
- TriggerSource_UserOutput0
 - CameraDefsC.h, [324](#)
- TriggerSource_UserOutput1
 - CameraDefsC.h, [324](#)
- TriggerSource_UserOutput2
 - CameraDefsC.h, [324](#)
- TriggerSource_UserOutput3
 - CameraDefsC.h, [324](#)
- True
 - SpinnakerDefsC.h, [451](#)
- type
 - _spinLibraryVersion, [202](#)
- UNKNOWN_PIXELFORMAT
 - CameraDefsC.h, [298](#)
- UnknownNode
 - SpinnakerGenApiDefsC.h, [501](#)
- Unsigned
 - SpinnakerGenApiDefsC.h, [501](#)
- USB

- SpinnakerGenApiDefsC.h, [502](#)
- UserOutputSelector
 - _quickSpin, [164](#)
- UserOutputSelector_UserOutput0
 - CameraDefsC.h, [324](#)
- UserOutputSelector_UserOutput1
 - CameraDefsC.h, [324](#)
- UserOutputSelector_UserOutput2
 - CameraDefsC.h, [324](#)
- UserOutputSelector_UserOutput3
 - CameraDefsC.h, [324](#)
- UserOutputValue
 - _quickSpin, [165](#)
- UserOutputValueAll
 - _quickSpin, [165](#)
- UserOutputValueAllMask
 - _quickSpin, [165](#)
- UserSetDefault
 - _quickSpin, [165](#)
- UserSetDefault_Default
 - CameraDefsC.h, [325](#)
- UserSetDefault_UserSet0
 - CameraDefsC.h, [325](#)
- UserSetDefault_UserSet1
 - CameraDefsC.h, [325](#)
- UserSetFeatureEnable
 - _quickSpin, [165](#)
- UserSetLoad
 - _quickSpin, [165](#)
- UserSetSave
 - _quickSpin, [165](#)
- UserSetSelector
 - _quickSpin, [165](#)
- UserSetSelector_Default
 - CameraDefsC.h, [325](#)
- UserSetSelector_UserSet0
 - CameraDefsC.h, [325](#)
- UserSetSelector_UserSet1
 - CameraDefsC.h, [325](#)
- V3_3Enable
 - _quickSpin, [166](#)
- ValueNode
 - SpinnakerGenApiDefsC.h, [500](#)
- Varying
 - SpinnakerGenApiDefsC.h, [502](#)
- WhiteClip
 - _quickSpin, [166](#)
- WhiteClipSelector
 - _quickSpin, [166](#)
- WhiteClipSelector_All
 - CameraDefsC.h, [325](#)
- WhiteClipSelector_Blue
 - CameraDefsC.h, [325](#)
- WhiteClipSelector_Green
 - CameraDefsC.h, [325](#)
- WhiteClipSelector_Red
 - CameraDefsC.h, [325](#)
- WhiteClipSelector_Tap1
 - CameraDefsC.h, [325](#)
- WhiteClipSelector_Tap2
 - CameraDefsC.h, [325](#)
- WhiteClipSelector_U
 - CameraDefsC.h, [325](#)
- WhiteClipSelector_V
 - CameraDefsC.h, [325](#)
- WhiteClipSelector_Y
 - CameraDefsC.h, [325](#)
- Width
 - _quickSpin, [166](#)
- width
 - _spinAVIOption, [191](#)
 - _spinH264Option, [199](#)
 - _spinMJPEGOption, [204](#)
- WidthMax
 - _quickSpin, [166](#)
- WO
 - SpinnakerGenApiDefsC.h, [497](#)
- WriteAround
 - SpinnakerGenApiDefsC.h, [497](#)
- WriteThrough
 - SpinnakerGenApiDefsC.h, [497](#)
- xvAll
 - SpinnakerGenApiDefsC.h, [503](#)
- xvCycles
 - SpinnakerGenApiDefsC.h, [503](#)
- xvDefault
 - SpinnakerGenApiDefsC.h, [503](#)
- xvLoad
 - SpinnakerGenApiDefsC.h, [503](#)
- xvSFNC
 - SpinnakerGenApiDefsC.h, [503](#)
- Yes
 - SpinnakerGenApiDefsC.h, [503](#)