

Spinnaker C

4.0.0.116

Generated by Doxygen 1.9.1

| | |
|--|-----------|
| 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 Spinnaker 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 Networking Best Practices | 15 |
| 8.1 GenTL Overview | 15 |
| 8.2 GenTL Overview | 15 |
| 9 Module Index | 17 |
| 9.1 Modules | 17 |
| 10 Data Structure Index | 19 |
| 10.1 Data Structures | 19 |
| 11 File Index | 21 |
| 11.1 File List | 21 |
| 12 Module Documentation | 23 |
| 12.1 Spinnaker C Definitions | 23 |
| 12.1.1 Detailed Description | 23 |
| 12.2 Camera Enumerations | 24 |
| 12.2.1 Detailed Description | 56 |
| 12.2.2 Enumeration Type Documentation | 56 |
| 12.2.2.1 spinAcquisitionModeEnums | 56 |

| | |
|---|----|
| 12.2.2.2 spinAcquisitionStatusSelectorEnums | 56 |
| 12.2.2.3 spinActionUnconditionalModeEnums | 57 |
| 12.2.2.4 spinAdcBitDepthEnums | 57 |
| 12.2.2.5 spinAutoAlgorithmSelectorEnums | 58 |
| 12.2.2.6 spinAutoExposureControlPriorityEnums | 58 |
| 12.2.2.7 spinAutoExposureLightingModeEnums | 58 |
| 12.2.2.8 spinAutoExposureMeteringModeEnums | 59 |
| 12.2.2.9 spinAutoExposureTargetGreyValueAutoEnums | 59 |
| 12.2.2.10 spinBalanceRatioSelectorEnums | 59 |
| 12.2.2.11 spinBalanceWhiteAutoEnums | 60 |
| 12.2.2.12 spinBalanceWhiteAutoProfileEnums | 60 |
| 12.2.2.13 spinBinningHorizontalModeEnums | 60 |
| 12.2.2.14 spinBinningSelectorEnums | 61 |
| 12.2.2.15 spinBinningVerticalModeEnums | 61 |
| 12.2.2.16 spinBlackLevelAutoBalanceEnums | 61 |
| 12.2.2.17 spinBlackLevelAutoEnums | 62 |
| 12.2.2.18 spinBlackLevelSelectorEnums | 62 |
| 12.2.2.19 spinChunkBlackLevelSelectorEnums | 62 |
| 12.2.2.20 spinChunkCounterSelectorEnums | 63 |
| 12.2.2.21 spinChunkEncoderSelectorEnums | 63 |
| 12.2.2.22 spinChunkEncoderStatusEnums | 63 |
| 12.2.2.23 spinChunkExposureTimeSelectorEnums | 64 |
| 12.2.2.24 spinChunkGainSelectorEnums | 64 |
| 12.2.2.25 spinChunkImageComponentEnums | 64 |
| 12.2.2.26 spinChunkPixelFormatEnums | 65 |
| 12.2.2.27 spinChunkRegionIDEnums | 65 |
| 12.2.2.28 spinChunkScan3dCoordinateReferenceSelectorEnums | 66 |
| 12.2.2.29 spinChunkScan3dCoordinateSelectorEnums | 66 |
| 12.2.2.30 spinChunkScan3dCoordinateSystemEnums | 66 |
| 12.2.2.31 spinChunkScan3dCoordinateSystemReferenceEnums | 67 |
| 12.2.2.32 spinChunkScan3dCoordinateTransformSelectorEnums | 67 |
| 12.2.2.33 spinChunkScan3dDistanceUnitEnums | 67 |
| 12.2.2.34 spinChunkScan3dOutputModeEnums | 68 |
| 12.2.2.35 spinChunkSelectorEnums | 69 |
| 12.2.2.36 spinChunkSourceIDEnums | 69 |
| 12.2.2.37 spinChunkTimerSelectorEnums | 69 |
| 12.2.2.38 spinChunkTransferStreamIDEnums | 70 |
| 12.2.2.39 spinCIConfigurationEnums | 70 |
| 12.2.2.40 spinCITimeSlotsCountEnums | 71 |
| 12.2.2.41 spinColorTransformationSelectorEnums | 71 |
| 12.2.2.42 spinColorTransformationValueSelectorEnums | 71 |
| 12.2.2.43 spinCompressionSaturationPriorityEnums | 72 |

| | |
|--|----|
| 12.2.2.44 spinCounterEventActivationEnums | 72 |
| 12.2.2.45 spinCounterEventSourceEnums | 72 |
| 12.2.2.46 spinCounterResetActivationEnums | 73 |
| 12.2.2.47 spinCounterResetSourceEnums | 73 |
| 12.2.2.48 spinCounterSelectorEnums | 74 |
| 12.2.2.49 spinCounterStatusEnums | 74 |
| 12.2.2.50 spinCounterTriggerActivationEnums | 75 |
| 12.2.2.51 spinCounterTriggerSourceEnums | 75 |
| 12.2.2.52 spinCxpConnectionTestModeEnums | 76 |
| 12.2.2.53 spinCxpLinkConfigurationEnums | 76 |
| 12.2.2.54 spinCxpLinkConfigurationPreferredEnums | 77 |
| 12.2.2.55 spinCxpLinkConfigurationStatusEnums | 78 |
| 12.2.2.56 spinCxpPoCxpStatusEnums | 79 |
| 12.2.2.57 spinDecimationHorizontalModeEnums | 79 |
| 12.2.2.58 spinDecimationSelectorEnums | 79 |
| 12.2.2.59 spinDecimationVerticalModeEnums | 80 |
| 12.2.2.60 spinDefectCorrectionModeEnums | 80 |
| 12.2.2.61 spinDeinterlacingEnums | 80 |
| 12.2.2.62 spinDeviceCharacterSetEnums | 81 |
| 12.2.2.63 spinDeviceClockSelectorEnums | 81 |
| 12.2.2.64 spinDeviceConnectionStatusEnums | 81 |
| 12.2.2.65 spinDeviceIndicatorModeEnums | 82 |
| 12.2.2.66 spinDeviceLinkHeartbeatModeEnums | 82 |
| 12.2.2.67 spinDeviceLinkThroughputLimitModeEnums | 82 |
| 12.2.2.68 spinDevicePowerSupplySelectorEnums | 83 |
| 12.2.2.69 spinDeviceRegistersEndiannessEnums | 83 |
| 12.2.2.70 spinDeviceScanTypeEnums | 83 |
| 12.2.2.71 spinDeviceSerialPortBaudRateEnums | 83 |
| 12.2.2.72 spinDeviceSerialPortSelectorEnums | 84 |
| 12.2.2.73 spinDeviceStreamChannelEndiannessEnums | 84 |
| 12.2.2.74 spinDeviceStreamChannelTypeEnums | 84 |
| 12.2.2.75 spinDeviceTapGeometryEnums | 85 |
| 12.2.2.76 spinDeviceTemperatureSelectorEnums | 86 |
| 12.2.2.77 spinDeviceTLTypeEnums | 86 |
| 12.2.2.78 spinDeviceTypeEnums | 87 |
| 12.2.2.79 spinEncoderModeEnums | 87 |
| 12.2.2.80 spinEncoderOutputModeEnums | 87 |
| 12.2.2.81 spinEncoderResetActivationEnums | 88 |
| 12.2.2.82 spinEncoderResetSourceEnums | 88 |
| 12.2.2.83 spinEncoderSelectorEnums | 89 |
| 12.2.2.84 spinEncoderSourceAEnums | 90 |
| 12.2.2.85 spinEncoderSourceBEnums | 90 |

| | |
|---|-----|
| 12.2.2.86 spinEncoderStatusEnums | 90 |
| 12.2.2.87 spinEventNotificationEnums | 91 |
| 12.2.2.88 spinEventSelectorEnums | 91 |
| 12.2.2.89 spinExposureActiveModeEnums | 91 |
| 12.2.2.90 spinExposureAutoEnums | 92 |
| 12.2.2.91 spinExposureModeEnums | 92 |
| 12.2.2.92 spinExposureTimeModeEnums | 92 |
| 12.2.2.93 spinExposureTimeSelectorEnums | 93 |
| 12.2.2.94 spinFileOpenModeEnums | 93 |
| 12.2.2.95 spinFileOperationSelectorEnums | 93 |
| 12.2.2.96 spinFileOperationStatusEnums | 94 |
| 12.2.2.97 spinFileSelectorEnums | 94 |
| 12.2.2.98 spinGainAutoBalanceEnums | 94 |
| 12.2.2.99 spinGainAutoEnums | 96 |
| 12.2.2.100 spinGainSelectorEnums | 96 |
| 12.2.2.101 spinGevCCPEnums | 96 |
| 12.2.2.102 spinGevCurrentPhysicalLinkConfigurationEnums | 97 |
| 12.2.2.103 spinGevGVCPExtendedStatusCodesSelectorEnums | 97 |
| 12.2.2.104 spinGevGVSPExtendedIDModeEnums | 97 |
| 12.2.2.105 spinGevIEEE1588ClockAccuracyEnums | 98 |
| 12.2.2.106 spinGevIEEE1588ModeEnums | 98 |
| 12.2.2.107 spinGevIEEE1588StatusEnums | 98 |
| 12.2.2.108 spinGevIPConfigurationStatusEnums | 99 |
| 12.2.2.109 spinGevPhysicalLinkConfigurationEnums | 99 |
| 12.2.2.110 spinGevSupportedOptionSelectorEnums | 100 |
| 12.2.2.111 spinImageComponentSelectorEnums | 100 |
| 12.2.2.112 spinImageCompressionJPEGFormatOptionEnums | 101 |
| 12.2.2.113 spinImageCompressionModeEnums | 101 |
| 12.2.2.114 spinImageCompressionRateOptionEnums | 102 |
| 12.2.2.115 spinLineFormatEnums | 102 |
| 12.2.2.116 spinLineInputFilterSelectorEnums | 102 |
| 12.2.2.117 spinLineModeEnums | 103 |
| 12.2.2.118 spinLineSelectorEnums | 103 |
| 12.2.2.119 spinLineSourceEnums | 103 |
| 12.2.2.120 spinLogicBlockLUTInputActivationEnums | 104 |
| 12.2.2.121 spinLogicBlockLUTInputSelectorEnums | 104 |
| 12.2.2.122 spinLogicBlockLUTInputSourceEnums | 105 |
| 12.2.2.123 spinLogicBlockLUTSelectorEnums | 105 |
| 12.2.2.124 spinLogicBlockSelectorEnums | 106 |
| 12.2.2.125 spinLUTSelectorEnums | 106 |
| 12.2.2.126 spinPixelColorFilterEnums | 106 |
| 12.2.2.127 spinPixelFormatEnums | 107 |

| | |
|---|-----|
| 12.2.2.128 spinPixelFormatInfoSelectorEnums | 112 |
| 12.2.2.129 spinPixelSizeEnums | 118 |
| 12.2.2.130 spinRegionDestinationEnums | 119 |
| 12.2.2.131 spinRegionModeEnums | 119 |
| 12.2.2.132 spinRegionSelectorEnums | 119 |
| 12.2.2.133 spinRgbTransformLightSourceEnums | 120 |
| 12.2.2.134 spinScan3dCoordinateReferenceSelectorEnums | 120 |
| 12.2.2.135 spinScan3dCoordinateSelectorEnums | 121 |
| 12.2.2.136 spinScan3dCoordinateSystemEnums | 121 |
| 12.2.2.137 spinScan3dCoordinateSystemReferenceEnums | 121 |
| 12.2.2.138 spinScan3dCoordinateTransformSelectorEnums | 122 |
| 12.2.2.139 spinScan3dDistanceUnitEnums | 122 |
| 12.2.2.140 spinScan3dOutputModeEnums | 122 |
| 12.2.2.141 spinSensorDigitizationTapsEnums | 124 |
| 12.2.2.142 spinSensorShutterModeEnums | 125 |
| 12.2.2.143 spinSensorTapsEnums | 125 |
| 12.2.2.144 spinSequencerConfigurationModeEnums | 126 |
| 12.2.2.145 spinSequencerConfigurationValidEnums | 126 |
| 12.2.2.146 spinSequencerModeEnums | 126 |
| 12.2.2.147 spinSequencerSetValidEnums | 126 |
| 12.2.2.148 spinSequencerTriggerActivationEnums | 127 |
| 12.2.2.149 spinSequencerTriggerSourceEnums | 127 |
| 12.2.2.150 spinSerialPortBaudRateEnums | 127 |
| 12.2.2.151 spinSerialPortParityEnums | 128 |
| 12.2.2.152 spinSerialPortSelectorEnums | 128 |
| 12.2.2.153 spinSerialPortSourceEnums | 129 |
| 12.2.2.154 spinSerialPortStopBitsEnums | 129 |
| 12.2.2.155 spinSoftwareSignalSelectorEnums | 129 |
| 12.2.2.156 spinSourceSelectorEnums | 130 |
| 12.2.2.157 spinTestPatternEnums | 130 |
| 12.2.2.158 spinTestPatternGeneratorSelectorEnums | 130 |
| 12.2.2.159 spinTimerSelectorEnums | 131 |
| 12.2.2.160 spinTimerStatusEnums | 131 |
| 12.2.2.161 spinTimerTriggerActivationEnums | 131 |
| 12.2.2.162 spinTimerTriggerSourceEnums | 132 |
| 12.2.2.163 spinTransferComponentSelectorEnums | 133 |
| 12.2.2.164 spinTransferControlModeEnums | 133 |
| 12.2.2.165 spinTransferOperationModeEnums | 134 |
| 12.2.2.166 spinTransferQueueModeEnums | 134 |
| 12.2.2.167 spinTransferSelectorEnums | 134 |
| 12.2.2.168 spinTransferStatusSelectorEnums | 135 |
| 12.2.2.169 spinTransferTriggerActivationEnums | 135 |

| | |
|---|-----|
| 12.2.2.170 spinTransferTriggerModeEnums | 135 |
| 12.2.2.171 spinTransferTriggerSelectorEnums | 136 |
| 12.2.2.172 spinTransferTriggerSourceEnums | 136 |
| 12.2.2.173 spinTriggerActivationEnums | 137 |
| 12.2.2.174 spinTriggerModeEnums | 138 |
| 12.2.2.175 spinTriggerOverlapEnums | 138 |
| 12.2.2.176 spinTriggerSelectorEnums | 138 |
| 12.2.2.177 spinTriggerSourceEnums | 139 |
| 12.2.2.178 spinUserOutputSelectorEnums | 139 |
| 12.2.2.179 spinUserSetDefaultEnums | 139 |
| 12.2.2.180 spinUserSetSelectorEnums | 140 |
| 12.2.2.181 spinWhiteClipSelectorEnums | 140 |
| 12.3 Chunk Data Structures | 141 |
| 12.3.1 Detailed Description | 141 |
| 12.4 Spinnaker C QuickSpin API | 141 |
| 12.4.1 Detailed Description | 142 |
| 12.5 QuickSpin Access | 142 |
| 12.6 Spinnaker C API | 142 |
| 12.6.1 Detailed Description | 142 |
| 12.7 Error Handling | 142 |
| 12.8 System Access | 143 |
| 12.9 InterfaceList Access | 143 |
| 12.10 CameraList Access | 143 |
| 12.11 ImageList Access | 143 |
| 12.12 Interface Access | 143 |
| 12.13 Camera Access | 143 |
| 12.14 Image Access | 144 |
| 12.15 Image Processor Access | 144 |
| 12.16 Event Access | 147 |
| 12.17 ImageStatistics Access | 147 |
| 12.18 Logging Event Data Access | 147 |
| 12.19 Device Event Data Access | 147 |
| 12.20 Chunk data access | 147 |
| 12.21 Spinnaker C Handles | 148 |
| 12.22 Spinnaker C Function Signatures | 148 |
| 12.23 Spinnaker C Enumerations | 148 |
| 12.24 Spinnaker C Structures | 148 |
| 12.25 Spinnaker C GenICam API | 148 |
| 12.26 Node Map Access | 148 |
| 12.27 Node Access | 148 |
| 12.28 IValue Access | 149 |
| 12.29 String Access | 149 |

| | |
|--|------------|
| 12.30 Integer Access | 149 |
| 12.31 IFloat Access | 149 |
| 12.32 IEnumeration Access | 149 |
| 12.33 IEnumEntry Access | 150 |
| 12.34 IBoolean Access | 150 |
| 12.35 ICommand Access | 150 |
| 12.36 ICategory Access | 150 |
| 12.37 IRegister Access | 150 |
| 12.38 Spinnaker C GenICam Handles | 150 |
| 12.39 Spinnaker C GenICam Enumerations | 151 |
| 12.40 SpinVideo Recording Access | 151 |
| 12.41 Transport Layer Enumerations | 151 |
| 12.41.1 Detailed Description | 153 |
| 12.41.2 Enumeration Type Documentation | 153 |
| 12.41.2.1 spinTLDeviceAccessStatusEnums | 153 |
| 12.41.2.2 spinTLDeviceCurrentSpeedEnums | 153 |
| 12.41.2.3 spinTLDeviceEndianessMechanismEnums | 154 |
| 12.41.2.4 spinTLDeviceTypeEnums | 154 |
| 12.41.2.5 spinTLFLIRFilterDriverStatusEnums | 154 |
| 12.41.2.6 spinTLGenICamXMLLocationEnums | 155 |
| 12.41.2.7 spinTLGevCCPEnums | 155 |
| 12.41.2.8 spinTLGUIXMLLocationEnums | 155 |
| 12.41.2.9 spinTLInterfaceTypeEnums | 156 |
| 12.41.2.10 spinTLPOEStatusEnums | 156 |
| 12.41.2.11 spinTLStreamBufferCountModeEnums | 156 |
| 12.41.2.12 spinTLStreamBufferHandlingModeEnums | 157 |
| 12.41.2.13 spinTLStreamModeEnums | 157 |
| 12.41.2.14 spinTLStreamTypeEnums | 158 |
| 12.41.2.15 spinTLTeledyneGigeVisionFilterDriverStatusEnums | 158 |
| 12.41.2.16 spinTLTLTypeEnums | 159 |
| 12.42 TLDevice Structures | 159 |
| 12.42.1 Detailed Description | 159 |
| 12.43 TLInterface Structures | 159 |
| 12.43.1 Detailed Description | 160 |
| 12.44 TLStream Structures | 160 |
| 12.44.1 Detailed Description | 160 |
| 12.45 TLSystem Structures | 160 |
| 12.45.1 Detailed Description | 160 |
| 13 Data Structure Documentation | 161 |
| 13.1 actionCommandResult Struct Reference | 161 |
| 13.1.1 Detailed Description | 161 |

| | |
|--|-----|
| 13.1.2 Field Documentation | 161 |
| 13.1.2.1 DeviceAddress | 161 |
| 13.1.2.2 Status | 161 |
| 13.2 quickSpin Struct Reference | 162 |
| 13.2.1 Field Documentation | 174 |
| 13.2.1.1 AasRoiEnable | 174 |
| 13.2.1.2 AasRoiHeight | 174 |
| 13.2.1.3 AasRoiOffsetX | 174 |
| 13.2.1.4 AasRoiOffsetY | 174 |
| 13.2.1.5 AasRoiWidth | 175 |
| 13.2.1.6 AcquisitionAbort | 175 |
| 13.2.1.7 AcquisitionArm | 175 |
| 13.2.1.8 AcquisitionBurstFrameCount | 175 |
| 13.2.1.9 AcquisitionFrameCount | 175 |
| 13.2.1.10 AcquisitionFrameRate | 175 |
| 13.2.1.11 AcquisitionFrameRateEnable | 175 |
| 13.2.1.12 AcquisitionLineRate | 175 |
| 13.2.1.13 AcquisitionMode | 176 |
| 13.2.1.14 AcquisitionResultingFrameRate | 176 |
| 13.2.1.15 AcquisitionStart | 176 |
| 13.2.1.16 AcquisitionStatus | 176 |
| 13.2.1.17 AcquisitionStatusSelector | 176 |
| 13.2.1.18 AcquisitionStop | 176 |
| 13.2.1.19 ActionDeviceKey | 176 |
| 13.2.1.20 ActionGroupKey | 176 |
| 13.2.1.21 ActionGroupMask | 177 |
| 13.2.1.22 ActionQueueSize | 177 |
| 13.2.1.23 ActionSelector | 177 |
| 13.2.1.24 ActionUnconditionalMode | 177 |
| 13.2.1.25 AdaptiveCompressionEnable | 177 |
| 13.2.1.26 AdcBitDepth | 177 |
| 13.2.1.27 aPAUSEMACCtrlFramesReceived | 177 |
| 13.2.1.28 aPAUSEMACCtrlFramesTransmitted | 177 |
| 13.2.1.29 AutoAlgorithmSelector | 178 |
| 13.2.1.30 AutoExposureControlLoopDamping | 178 |
| 13.2.1.31 AutoExposureControlPriority | 178 |
| 13.2.1.32 AutoExposureEVCompensation | 178 |
| 13.2.1.33 AutoExposureExposureTimeLowerLimit | 178 |
| 13.2.1.34 AutoExposureExposureTimeUpperLimit | 178 |
| 13.2.1.35 AutoExposureGainLowerLimit | 178 |
| 13.2.1.36 AutoExposureGainUpperLimit | 178 |
| 13.2.1.37 AutoExposureGreyValueLowerLimit | 179 |

| | |
|---|-----|
| 13.2.1.38 AutoExposureGreyValueUpperLimit | 179 |
| 13.2.1.39 AutoExposureLightingMode | 179 |
| 13.2.1.40 AutoExposureMeteringMode | 179 |
| 13.2.1.41 AutoExposureTargetGreyValue | 179 |
| 13.2.1.42 AutoExposureTargetGreyValueAuto | 179 |
| 13.2.1.43 BalanceRatio | 179 |
| 13.2.1.44 BalanceRatioSelector | 179 |
| 13.2.1.45 BalanceWhiteAuto | 180 |
| 13.2.1.46 BalanceWhiteAutoDamping | 180 |
| 13.2.1.47 BalanceWhiteAutoLowerLimit | 180 |
| 13.2.1.48 BalanceWhiteAutoProfile | 180 |
| 13.2.1.49 BalanceWhiteAutoUpperLimit | 180 |
| 13.2.1.50 BinningHorizontal | 180 |
| 13.2.1.51 BinningHorizontalMode | 180 |
| 13.2.1.52 BinningSelector | 180 |
| 13.2.1.53 BinningVertical | 181 |
| 13.2.1.54 BinningVerticalMode | 181 |
| 13.2.1.55 BlackLevel | 181 |
| 13.2.1.56 BlackLevelAuto | 181 |
| 13.2.1.57 BlackLevelAutoBalance | 181 |
| 13.2.1.58 BlackLevelClampingEnable | 181 |
| 13.2.1.59 BlackLevelRaw | 181 |
| 13.2.1.60 BlackLevelSelector | 181 |
| 13.2.1.61 ChunkBlackLevel | 182 |
| 13.2.1.62 ChunkBlackLevelSelector | 182 |
| 13.2.1.63 ChunkCompressionMode | 182 |
| 13.2.1.64 ChunkCompressionRatio | 182 |
| 13.2.1.65 ChunkCounterSelector | 182 |
| 13.2.1.66 ChunkCounterValue | 182 |
| 13.2.1.67 ChunkCRC | 182 |
| 13.2.1.68 ChunkEnable | 182 |
| 13.2.1.69 ChunkEncoderSelector | 183 |
| 13.2.1.70 ChunkEncoderStatus | 183 |
| 13.2.1.71 ChunkEncoderValue | 183 |
| 13.2.1.72 ChunkExposureEndLineStatusAll | 183 |
| 13.2.1.73 ChunkExposureTime | 183 |
| 13.2.1.74 ChunkExposureTimeSelector | 183 |
| 13.2.1.75 ChunkFrameID | 183 |
| 13.2.1.76 ChunkGain | 183 |
| 13.2.1.77 ChunkGainSelector | 184 |
| 13.2.1.78 ChunkHeight | 184 |
| 13.2.1.79 ChunkImage | 184 |

| | |
|---|-----|
| 13.2.1.80 ChunkImageComponent | 184 |
| 13.2.1.81 ChunkInferenceBoundingBoxResult | 184 |
| 13.2.1.82 ChunkInferenceConfidence | 184 |
| 13.2.1.83 ChunkInferenceFrameId | 184 |
| 13.2.1.84 ChunkInferenceResult | 184 |
| 13.2.1.85 ChunkLinePitch | 185 |
| 13.2.1.86 ChunkLineStatusAll | 185 |
| 13.2.1.87 ChunkModeActive | 185 |
| 13.2.1.88 ChunkOffsetX | 185 |
| 13.2.1.89 ChunkOffsetY | 185 |
| 13.2.1.90 ChunkPartSelector | 185 |
| 13.2.1.91 ChunkPixelDynamicRangeMax | 185 |
| 13.2.1.92 ChunkPixelDynamicRangeMin | 185 |
| 13.2.1.93 ChunkPixelFormat | 186 |
| 13.2.1.94 ChunkRegionID | 186 |
| 13.2.1.95 ChunkScan3dAxisMax | 186 |
| 13.2.1.96 ChunkScan3dAxisMin | 186 |
| 13.2.1.97 ChunkScan3dCoordinateOffset | 186 |
| 13.2.1.98 ChunkScan3dCoordinateReferenceSelector | 186 |
| 13.2.1.99 ChunkScan3dCoordinateReferenceValue | 186 |
| 13.2.1.100 ChunkScan3dCoordinateScale | 186 |
| 13.2.1.101 ChunkScan3dCoordinateSelector | 187 |
| 13.2.1.102 ChunkScan3dCoordinateSystem | 187 |
| 13.2.1.103 ChunkScan3dCoordinateSystemReference | 187 |
| 13.2.1.104 ChunkScan3dCoordinateTransformSelector | 187 |
| 13.2.1.105 ChunkScan3dDistanceUnit | 187 |
| 13.2.1.106 ChunkScan3dInvalidDataFlag | 187 |
| 13.2.1.107 ChunkScan3dInvalidDataValue | 187 |
| 13.2.1.108 ChunkScan3dOutputMode | 187 |
| 13.2.1.109 ChunkScan3dTransformValue | 188 |
| 13.2.1.110 ChunkScanLineSelector | 188 |
| 13.2.1.111 ChunkSelector | 188 |
| 13.2.1.112 ChunkSequencerSetActive | 188 |
| 13.2.1.113 ChunkSerialData | 188 |
| 13.2.1.114 ChunkSerialDataLength | 188 |
| 13.2.1.115 ChunkSerialReceiveOverflow | 188 |
| 13.2.1.116 ChunkSourceID | 188 |
| 13.2.1.117 ChunkStreamChannelID | 189 |
| 13.2.1.118 ChunkTimerSelector | 189 |
| 13.2.1.119 ChunkTimerValue | 189 |
| 13.2.1.120 ChunkTimestamp | 189 |
| 13.2.1.121 ChunkTimestampLatchValue | 189 |

| | |
|--|-----|
| 13.2.1.122 ChunkTransferBlockID | 189 |
| 13.2.1.123 ChunkTransferQueueCurrentBlockCount | 189 |
| 13.2.1.124 ChunkTransferStreamID | 189 |
| 13.2.1.125 ChunkWidth | 190 |
| 13.2.1.126 CIConfiguration | 190 |
| 13.2.1.127 CITimeSlotsCount | 190 |
| 13.2.1.128 ColorTransformationEnable | 190 |
| 13.2.1.129 ColorTransformationSelector | 190 |
| 13.2.1.130 ColorTransformationValue | 190 |
| 13.2.1.131 ColorTransformationValueSelector | 190 |
| 13.2.1.132 CompressionRatio | 190 |
| 13.2.1.133 CompressionSaturationPriority | 191 |
| 13.2.1.134 CounterDelay | 191 |
| 13.2.1.135 CounterDuration | 191 |
| 13.2.1.136 CounterEventActivation | 191 |
| 13.2.1.137 CounterEventSource | 191 |
| 13.2.1.138 CounterReset | 191 |
| 13.2.1.139 CounterResetActivation | 191 |
| 13.2.1.140 CounterResetSource | 191 |
| 13.2.1.141 CounterSelector | 192 |
| 13.2.1.142 CounterStatus | 192 |
| 13.2.1.143 CounterTriggerActivation | 192 |
| 13.2.1.144 CounterTriggerSource | 192 |
| 13.2.1.145 CounterValue | 192 |
| 13.2.1.146 CounterValueAtReset | 192 |
| 13.2.1.147 CxpConnectionSelector | 192 |
| 13.2.1.148 CxpConnectionTestErrorCount | 192 |
| 13.2.1.149 CxpConnectionTestMode | 193 |
| 13.2.1.150 CxpConnectionTestPacketCount | 193 |
| 13.2.1.151 CxpLinkConfiguration | 193 |
| 13.2.1.152 CxpLinkConfigurationPreferred | 193 |
| 13.2.1.153 CxpLinkConfigurationStatus | 193 |
| 13.2.1.154 CxpPoCxpAuto | 193 |
| 13.2.1.155 CxpPoCxpStatus | 193 |
| 13.2.1.156 CxpPoCxpTripReset | 193 |
| 13.2.1.157 CxpPoCxpTurnOff | 194 |
| 13.2.1.158 DecimationHorizontal | 194 |
| 13.2.1.159 DecimationHorizontalMode | 194 |
| 13.2.1.160 DecimationSelector | 194 |
| 13.2.1.161 DecimationVertical | 194 |
| 13.2.1.162 DecimationVerticalMode | 194 |
| 13.2.1.163 DefectCorrectionMode | 194 |

| | |
|---|-----|
| 13.2.1.164 DefectCorrectStaticEnable | 194 |
| 13.2.1.165 DefectTableApply | 195 |
| 13.2.1.166 DefectTableCoordinateX | 195 |
| 13.2.1.167 DefectTableCoordinateY | 195 |
| 13.2.1.168 DefectTableFactoryRestore | 195 |
| 13.2.1.169 DefectTableIndex | 195 |
| 13.2.1.170 DefectTablePixelCount | 195 |
| 13.2.1.171 DefectTableSave | 195 |
| 13.2.1.172 Deinterlacing | 195 |
| 13.2.1.173 DeviceCharacterSet | 196 |
| 13.2.1.174 DeviceClockFrequency | 196 |
| 13.2.1.175 DeviceClockSelector | 196 |
| 13.2.1.176 DeviceConnectionSelector | 196 |
| 13.2.1.177 DeviceConnectionSpeed | 196 |
| 13.2.1.178 DeviceConnectionStatus | 196 |
| 13.2.1.179 DeviceEventChannelCount | 196 |
| 13.2.1.180 DeviceFamilyName | 196 |
| 13.2.1.181 DeviceFeaturePersistenceEnd | 197 |
| 13.2.1.182 DeviceFeaturePersistenceStart | 197 |
| 13.2.1.183 DeviceFirmwareVersion | 197 |
| 13.2.1.184 DeviceGenCPVersionMajor | 197 |
| 13.2.1.185 DeviceGenCPVersionMinor | 197 |
| 13.2.1.186 DeviceID | 197 |
| 13.2.1.187 DeviceIndicatorMode | 197 |
| 13.2.1.188 DeviceLinkBandwidthReserve | 197 |
| 13.2.1.189 DeviceLinkCommandTimeout | 198 |
| 13.2.1.190 DeviceLinkConnectionCount | 198 |
| 13.2.1.191 DeviceLinkCurrentThroughput | 198 |
| 13.2.1.192 DeviceLinkHeartbeatMode | 198 |
| 13.2.1.193 DeviceLinkHeartbeatTimeout | 198 |
| 13.2.1.194 DeviceLinkSelector | 198 |
| 13.2.1.195 DeviceLinkSpeed | 198 |
| 13.2.1.196 DeviceLinkThroughputLimit | 198 |
| 13.2.1.197 DeviceLinkThroughputLimitMode | 199 |
| 13.2.1.198 DeviceManifestEntrySelector | 199 |
| 13.2.1.199 DeviceManifestPrimaryURL | 199 |
| 13.2.1.200 DeviceManifestSchemaMajorVersion | 199 |
| 13.2.1.201 DeviceManifestSchemaMinorVersion | 199 |
| 13.2.1.202 DeviceManifestSecondaryURL | 199 |
| 13.2.1.203 DeviceManifestXMLMajorVersion | 199 |
| 13.2.1.204 DeviceManifestXMLMinorVersion | 199 |
| 13.2.1.205 DeviceManifestXMLSubMinorVersion | 200 |

| | |
|--|-----|
| 13.2.1.206 DeviceManufacturerInfo | 200 |
| 13.2.1.207 DeviceMaxThroughput | 200 |
| 13.2.1.208 DeviceModelName | 200 |
| 13.2.1.209 DevicePowerSupplySelector | 200 |
| 13.2.1.210 DeviceRegistersCheck | 200 |
| 13.2.1.211 DeviceRegistersEndianness | 200 |
| 13.2.1.212 DeviceRegistersStreamingEnd | 200 |
| 13.2.1.213 DeviceRegistersStreamingStart | 201 |
| 13.2.1.214 DeviceRegistersValid | 201 |
| 13.2.1.215 DeviceReset | 201 |
| 13.2.1.216 DeviceScanType | 201 |
| 13.2.1.217 DeviceSerialNumber | 201 |
| 13.2.1.218 DeviceSerialPortBaudRate | 201 |
| 13.2.1.219 DeviceSerialPortSelector | 201 |
| 13.2.1.220 DeviceSFNCVersionMajor | 201 |
| 13.2.1.221 DeviceSFNCVersionMinor | 202 |
| 13.2.1.222 DeviceSFNCVersionSubMinor | 202 |
| 13.2.1.223 DeviceStreamChannelCount | 202 |
| 13.2.1.224 DeviceStreamChannelEndianness | 202 |
| 13.2.1.225 DeviceStreamChannelLink | 202 |
| 13.2.1.226 DeviceStreamChannelPacketSize | 202 |
| 13.2.1.227 DeviceStreamChannelSelector | 202 |
| 13.2.1.228 DeviceStreamChannelType | 202 |
| 13.2.1.229 DeviceTapGeometry | 203 |
| 13.2.1.230 DeviceTemperature | 203 |
| 13.2.1.231 DeviceTemperatureSelector | 203 |
| 13.2.1.232 DeviceTLType | 203 |
| 13.2.1.233 DeviceTLVersionMajor | 203 |
| 13.2.1.234 DeviceTLVersionMinor | 203 |
| 13.2.1.235 DeviceTLVersionSubMinor | 203 |
| 13.2.1.236 DeviceType | 203 |
| 13.2.1.237 DeviceUptime | 204 |
| 13.2.1.238 DeviceUserID | 204 |
| 13.2.1.239 DeviceVendorName | 204 |
| 13.2.1.240 DeviceVersion | 204 |
| 13.2.1.241 EncoderDivider | 204 |
| 13.2.1.242 EncoderMode | 204 |
| 13.2.1.243 EncoderOutputMode | 204 |
| 13.2.1.244 EncoderReset | 204 |
| 13.2.1.245 EncoderResetActivation | 205 |
| 13.2.1.246 EncoderResetSource | 205 |
| 13.2.1.247 EncoderSelector | 205 |

| | |
|---|-----|
| 13.2.1.248 EncoderSourceA | 205 |
| 13.2.1.249 EncoderSourceB | 205 |
| 13.2.1.250 EncoderStatus | 205 |
| 13.2.1.251 EncoderTimeout | 205 |
| 13.2.1.252 EncoderValue | 205 |
| 13.2.1.253 EncoderValueAtReset | 206 |
| 13.2.1.254 EnumerationCount | 206 |
| 13.2.1.255 EventAcquisitionEnd | 206 |
| 13.2.1.256 EventAcquisitionEndFrameID | 206 |
| 13.2.1.257 EventAcquisitionEndTimestamp | 206 |
| 13.2.1.258 EventAcquisitionError | 206 |
| 13.2.1.259 EventAcquisitionErrorFrameID | 206 |
| 13.2.1.260 EventAcquisitionErrorTimestamp | 206 |
| 13.2.1.261 EventAcquisitionStart | 207 |
| 13.2.1.262 EventAcquisitionStartFrameID | 207 |
| 13.2.1.263 EventAcquisitionStartTimestamp | 207 |
| 13.2.1.264 EventAcquisitionTransferEnd | 207 |
| 13.2.1.265 EventAcquisitionTransferEndFrameID | 207 |
| 13.2.1.266 EventAcquisitionTransferEndTimestamp | 207 |
| 13.2.1.267 EventAcquisitionTransferStart | 207 |
| 13.2.1.268 EventAcquisitionTransferStartFrameID | 207 |
| 13.2.1.269 EventAcquisitionTransferStartTimestamp | 208 |
| 13.2.1.270 EventAcquisitionTrigger | 208 |
| 13.2.1.271 EventAcquisitionTriggerFrameID | 208 |
| 13.2.1.272 EventAcquisitionTriggerTimestamp | 208 |
| 13.2.1.273 EventActionLate | 208 |
| 13.2.1.274 EventActionLateFrameID | 208 |
| 13.2.1.275 EventActionLateTimestamp | 208 |
| 13.2.1.276 EventCounter0End | 208 |
| 13.2.1.277 EventCounter0EndFrameID | 209 |
| 13.2.1.278 EventCounter0EndTimestamp | 209 |
| 13.2.1.279 EventCounter0Start | 209 |
| 13.2.1.280 EventCounter0StartFrameID | 209 |
| 13.2.1.281 EventCounter0StartTimestamp | 209 |
| 13.2.1.282 EventCounter1End | 209 |
| 13.2.1.283 EventCounter1EndFrameID | 209 |
| 13.2.1.284 EventCounter1EndTimestamp | 209 |
| 13.2.1.285 EventCounter1Start | 210 |
| 13.2.1.286 EventCounter1StartFrameID | 210 |
| 13.2.1.287 EventCounter1StartTimestamp | 210 |
| 13.2.1.288 EventEncoder0Restarted | 210 |
| 13.2.1.289 EventEncoder0RestartedFrameID | 210 |

| | |
|---|-----|
| 13.2.1.290 EventEncoder0RestartedTimestamp | 210 |
| 13.2.1.291 EventEncoder0Stopped | 210 |
| 13.2.1.292 EventEncoder0StoppedFrameID | 210 |
| 13.2.1.293 EventEncoder0StoppedTimestamp | 211 |
| 13.2.1.294 EventEncoder1Restarted | 211 |
| 13.2.1.295 EventEncoder1RestartedFrameID | 211 |
| 13.2.1.296 EventEncoder1RestartedTimestamp | 211 |
| 13.2.1.297 EventEncoder1Stopped | 211 |
| 13.2.1.298 EventEncoder1StoppedFrameID | 211 |
| 13.2.1.299 EventEncoder1StoppedTimestamp | 211 |
| 13.2.1.300 EventError | 211 |
| 13.2.1.301 EventErrorCode | 212 |
| 13.2.1.302 EventErrorFrameID | 212 |
| 13.2.1.303 EventErrorTimestamp | 212 |
| 13.2.1.304 EventExposureEnd | 212 |
| 13.2.1.305 EventExposureEndFrameID | 212 |
| 13.2.1.306 EventExposureEndTimestamp | 212 |
| 13.2.1.307 EventExposureStart | 212 |
| 13.2.1.308 EventExposureStartFrameID | 212 |
| 13.2.1.309 EventExposureStartTimestamp | 213 |
| 13.2.1.310 EventFrameBurstEnd | 213 |
| 13.2.1.311 EventFrameBurstEndFrameID | 213 |
| 13.2.1.312 EventFrameBurstEndTimestamp | 213 |
| 13.2.1.313 EventFrameBurstStart | 213 |
| 13.2.1.314 EventFrameBurstStartFrameID | 213 |
| 13.2.1.315 EventFrameBurstStartTimestamp | 213 |
| 13.2.1.316 EventFrameEnd | 213 |
| 13.2.1.317 EventFrameEndFrameID | 214 |
| 13.2.1.318 EventFrameEndTimestamp | 214 |
| 13.2.1.319 EventFrameStart | 214 |
| 13.2.1.320 EventFrameStartFrameID | 214 |
| 13.2.1.321 EventFrameStartTimestamp | 214 |
| 13.2.1.322 EventFrameTransferEnd | 214 |
| 13.2.1.323 EventFrameTransferEndFrameID | 214 |
| 13.2.1.324 EventFrameTransferEndTimestamp | 214 |
| 13.2.1.325 EventFrameTransferStart | 215 |
| 13.2.1.326 EventFrameTransferStartFrameID | 215 |
| 13.2.1.327 EventFrameTransferStartTimestamp | 215 |
| 13.2.1.328 EventFrameTrigger | 215 |
| 13.2.1.329 EventFrameTriggerFrameID | 215 |
| 13.2.1.330 EventFrameTriggerTimestamp | 215 |
| 13.2.1.331 EventLine0AnyEdge | 215 |

| | |
|--|-----|
| 13.2.1.332 EventLine0AnyEdgeFrameID | 215 |
| 13.2.1.333 EventLine0AnyEdgeTimestamp | 216 |
| 13.2.1.334 EventLine0FallingEdge | 216 |
| 13.2.1.335 EventLine0FallingEdgeFrameID | 216 |
| 13.2.1.336 EventLine0FallingEdgeTimestamp | 216 |
| 13.2.1.337 EventLine0RisingEdge | 216 |
| 13.2.1.338 EventLine0RisingEdgeFrameID | 216 |
| 13.2.1.339 EventLine0RisingEdgeTimestamp | 216 |
| 13.2.1.340 EventLine1AnyEdge | 216 |
| 13.2.1.341 EventLine1AnyEdgeFrameID | 217 |
| 13.2.1.342 EventLine1AnyEdgeTimestamp | 217 |
| 13.2.1.343 EventLine1FallingEdge | 217 |
| 13.2.1.344 EventLine1FallingEdgeFrameID | 217 |
| 13.2.1.345 EventLine1FallingEdgeTimestamp | 217 |
| 13.2.1.346 EventLine1RisingEdge | 217 |
| 13.2.1.347 EventLine1RisingEdgeFrameID | 217 |
| 13.2.1.348 EventLine1RisingEdgeTimestamp | 217 |
| 13.2.1.349 EventLinkSpeedChange | 218 |
| 13.2.1.350 EventLinkSpeedChangeFrameID | 218 |
| 13.2.1.351 EventLinkSpeedChangeTimestamp | 218 |
| 13.2.1.352 EventLinkTrigger0 | 218 |
| 13.2.1.353 EventLinkTrigger0FrameID | 218 |
| 13.2.1.354 EventLinkTrigger0Timestamp | 218 |
| 13.2.1.355 EventLinkTrigger1 | 218 |
| 13.2.1.356 EventLinkTrigger1FrameID | 218 |
| 13.2.1.357 EventLinkTrigger1Timestamp | 219 |
| 13.2.1.358 EventNotification | 219 |
| 13.2.1.359 EventSelector | 219 |
| 13.2.1.360 EventSequencerSetChange | 219 |
| 13.2.1.361 EventSequencerSetChangeFrameID | 219 |
| 13.2.1.362 EventSequencerSetChangeTimestamp | 219 |
| 13.2.1.363 EventSerialData | 219 |
| 13.2.1.364 EventSerialDataLength | 219 |
| 13.2.1.365 EventSerialPortReceive | 220 |
| 13.2.1.366 EventSerialPortReceiveTimestamp | 220 |
| 13.2.1.367 EventSerialReceiveOverflow | 220 |
| 13.2.1.368 EventStream0TransferBlockEnd | 220 |
| 13.2.1.369 EventStream0TransferBlockEndFrameID | 220 |
| 13.2.1.370 EventStream0TransferBlockEndTimestamp | 220 |
| 13.2.1.371 EventStream0TransferBlockStart | 220 |
| 13.2.1.372 EventStream0TransferBlockStartFrameID | 220 |
| 13.2.1.373 EventStream0TransferBlockStartTimestamp | 221 |

| | |
|--|-----|
| 13.2.1.374 EventStream0TransferBlockTrigger | 221 |
| 13.2.1.375 EventStream0TransferBlockTriggerFrameID | 221 |
| 13.2.1.376 EventStream0TransferBlockTriggerTimestamp | 221 |
| 13.2.1.377 EventStream0TransferBurstEnd | 221 |
| 13.2.1.378 EventStream0TransferBurstEndFrameID | 221 |
| 13.2.1.379 EventStream0TransferBurstEndTimestamp | 221 |
| 13.2.1.380 EventStream0TransferBurstStart | 221 |
| 13.2.1.381 EventStream0TransferBurstStartFrameID | 222 |
| 13.2.1.382 EventStream0TransferBurstStartTimestamp | 222 |
| 13.2.1.383 EventStream0TransferEnd | 222 |
| 13.2.1.384 EventStream0TransferEndFrameID | 222 |
| 13.2.1.385 EventStream0TransferEndTimestamp | 222 |
| 13.2.1.386 EventStream0TransferOverflow | 222 |
| 13.2.1.387 EventStream0TransferOverflowFrameID | 222 |
| 13.2.1.388 EventStream0TransferOverflowTimestamp | 222 |
| 13.2.1.389 EventStream0TransferPause | 223 |
| 13.2.1.390 EventStream0TransferPauseFrameID | 223 |
| 13.2.1.391 EventStream0TransferPauseTimestamp | 223 |
| 13.2.1.392 EventStream0TransferResume | 223 |
| 13.2.1.393 EventStream0TransferResumeFrameID | 223 |
| 13.2.1.394 EventStream0TransferResumeTimestamp | 223 |
| 13.2.1.395 EventStream0TransferStart | 223 |
| 13.2.1.396 EventStream0TransferStartFrameID | 223 |
| 13.2.1.397 EventStream0TransferStartTimestamp | 224 |
| 13.2.1.398 EventTest | 224 |
| 13.2.1.399 EventTestTimestamp | 224 |
| 13.2.1.400 EventTimer0End | 224 |
| 13.2.1.401 EventTimer0EndFrameID | 224 |
| 13.2.1.402 EventTimer0EndTimestamp | 224 |
| 13.2.1.403 EventTimer0Start | 224 |
| 13.2.1.404 EventTimer0StartFrameID | 224 |
| 13.2.1.405 EventTimer0StartTimestamp | 225 |
| 13.2.1.406 EventTimer1End | 225 |
| 13.2.1.407 EventTimer1EndFrameID | 225 |
| 13.2.1.408 EventTimer1EndTimestamp | 225 |
| 13.2.1.409 EventTimer1Start | 225 |
| 13.2.1.410 EventTimer1StartFrameID | 225 |
| 13.2.1.411 EventTimer1StartTimestamp | 225 |
| 13.2.1.412 ExposureActiveMode | 225 |
| 13.2.1.413 ExposureAuto | 226 |
| 13.2.1.414 ExposureMode | 226 |
| 13.2.1.415 ExposureTime | 226 |

| | |
|--|-----|
| 13.2.1.416 ExposureTimeMode | 226 |
| 13.2.1.417 ExposureTimeSelector | 226 |
| 13.2.1.418 FactoryReset | 226 |
| 13.2.1.419 FileAccessBuffer | 226 |
| 13.2.1.420 FileAccessLength | 226 |
| 13.2.1.421 FileAccessOffset | 227 |
| 13.2.1.422 FileOpenMode | 227 |
| 13.2.1.423 FileOperationExecute | 227 |
| 13.2.1.424 FileOperationResult | 227 |
| 13.2.1.425 FileOperationSelector | 227 |
| 13.2.1.426 FileOperationStatus | 227 |
| 13.2.1.427 FileSelector | 227 |
| 13.2.1.428 FileSize | 227 |
| 13.2.1.429 Gain | 228 |
| 13.2.1.430 GainAuto | 228 |
| 13.2.1.431 GainAutoBalance | 228 |
| 13.2.1.432 GainSelector | 228 |
| 13.2.1.433 Gamma | 228 |
| 13.2.1.434 GammaEnable | 228 |
| 13.2.1.435 GevActiveLinkCount | 228 |
| 13.2.1.436 GevCCP | 228 |
| 13.2.1.437 GevCurrentDefaultGateway | 229 |
| 13.2.1.438 GevCurrentIPAddress | 229 |
| 13.2.1.439 GevCurrentIPConfigurationDHCP | 229 |
| 13.2.1.440 GevCurrentIPConfigurationLLA | 229 |
| 13.2.1.441 GevCurrentIPConfigurationPersistentIP | 229 |
| 13.2.1.442 GevCurrentPhysicalLinkConfiguration | 229 |
| 13.2.1.443 GevCurrentSubnetMask | 229 |
| 13.2.1.444 GevDiscoveryAckDelay | 229 |
| 13.2.1.445 GevFirstURL | 230 |
| 13.2.1.446 GevGVCPExtendedStatusCodes | 230 |
| 13.2.1.447 GevGVCPExtendedStatusCodesSelector | 230 |
| 13.2.1.448 GevGVCPHeartbeatDisable | 230 |
| 13.2.1.449 GevGVCPPendingAck | 230 |
| 13.2.1.450 GevGVCPPendingTimeout | 230 |
| 13.2.1.451 GevGVSPExtendedIDMode | 230 |
| 13.2.1.452 GevHeartbeatTimeout | 230 |
| 13.2.1.453 GevIEEE1588 | 231 |
| 13.2.1.454 GevIEEE1588ClockAccuracy | 231 |
| 13.2.1.455 GevIEEE1588Mode | 231 |
| 13.2.1.456 GevIEEE1588Status | 231 |
| 13.2.1.457 GevInterfaceSelector | 231 |

| | | |
|------------|------------------------------------|-----|
| 13.2.1.458 | GevIPConfigurationStatus | 231 |
| 13.2.1.459 | GevMACAddress | 231 |
| 13.2.1.460 | GevMCDA | 231 |
| 13.2.1.461 | GevMCPHostPort | 232 |
| 13.2.1.462 | GevMCRC | 232 |
| 13.2.1.463 | GevMCSP | 232 |
| 13.2.1.464 | GevMCTT | 232 |
| 13.2.1.465 | GevNumberOfInterfaces | 232 |
| 13.2.1.466 | GevPAUSEFrameReception | 232 |
| 13.2.1.467 | GevPAUSEFrameTransmission | 232 |
| 13.2.1.468 | GevPersistentDefaultGateway | 232 |
| 13.2.1.469 | GevPersistentIPAddress | 233 |
| 13.2.1.470 | GevPersistentSubnetMask | 233 |
| 13.2.1.471 | GevPhysicalLinkConfiguration | 233 |
| 13.2.1.472 | GevPrimaryApplicationIPAddress | 233 |
| 13.2.1.473 | GevPrimaryApplicationSocket | 233 |
| 13.2.1.474 | GevPrimaryApplicationSwitchoverKey | 233 |
| 13.2.1.475 | GevSCCFGAllInTransmission | 233 |
| 13.2.1.476 | GevSCCFGExtendedChunkData | 233 |
| 13.2.1.477 | GevSCCFGPacketResendDestination | 234 |
| 13.2.1.478 | GevSCCFGUnconditionalStreaming | 234 |
| 13.2.1.479 | GevSCDA | 234 |
| 13.2.1.480 | GevSCPD | 234 |
| 13.2.1.481 | GevSCPDDirection | 234 |
| 13.2.1.482 | GevSCPHostPort | 234 |
| 13.2.1.483 | GevSCPIInterfaceIndex | 234 |
| 13.2.1.484 | GevSCPSBigEndian | 234 |
| 13.2.1.485 | GevSCPSDoNotFragment | 235 |
| 13.2.1.486 | GevSCPSFireTestPacket | 235 |
| 13.2.1.487 | GevSCPSPacketSize | 235 |
| 13.2.1.488 | GevSCSP | 235 |
| 13.2.1.489 | GevSCZoneConfigurationLock | 235 |
| 13.2.1.490 | GevSCZoneCount | 235 |
| 13.2.1.491 | GevSCZoneDirectionAll | 235 |
| 13.2.1.492 | GevSecondURL | 235 |
| 13.2.1.493 | GevStreamChannelSelector | 236 |
| 13.2.1.494 | GevSupportedOption | 236 |
| 13.2.1.495 | GevSupportedOptionSelector | 236 |
| 13.2.1.496 | GevTimestampTickFrequency | 236 |
| 13.2.1.497 | GuiXmlManifestAddress | 236 |
| 13.2.1.498 | Height | 236 |
| 13.2.1.499 | HeightMax | 236 |

| | |
|---|-----|
| 13.2.1.500 ImageComponentEnable | 236 |
| 13.2.1.501 ImageComponentSelector | 237 |
| 13.2.1.502 ImageCompressionBitrate | 237 |
| 13.2.1.503 ImageCompressionJPEGFormatOption | 237 |
| 13.2.1.504 ImageCompressionMode | 237 |
| 13.2.1.505 ImageCompressionQuality | 237 |
| 13.2.1.506 ImageCompressionRateOption | 237 |
| 13.2.1.507 IspEnable | 237 |
| 13.2.1.508 LineFilterWidth | 237 |
| 13.2.1.509 LineFormat | 238 |
| 13.2.1.510 LineInputFilterSelector | 238 |
| 13.2.1.511 LineInverter | 238 |
| 13.2.1.512 LineMode | 238 |
| 13.2.1.513 LinePitch | 238 |
| 13.2.1.514 LineSelector | 238 |
| 13.2.1.515 LineSource | 238 |
| 13.2.1.516 LineStatus | 238 |
| 13.2.1.517 LineStatusAll | 239 |
| 13.2.1.518 LinkErrorCount | 239 |
| 13.2.1.519 LinkUptime | 239 |
| 13.2.1.520 LogicBlockLUTInputActivation | 239 |
| 13.2.1.521 LogicBlockLUTInputSelector | 239 |
| 13.2.1.522 LogicBlockLUTInputSource | 239 |
| 13.2.1.523 LogicBlockLUTOutputValue | 239 |
| 13.2.1.524 LogicBlockLUTOutputValueAll | 239 |
| 13.2.1.525 LogicBlockLUTRowIndex | 240 |
| 13.2.1.526 LogicBlockLUTSelector | 240 |
| 13.2.1.527 LogicBlockSelector | 240 |
| 13.2.1.528 LUTEnable | 240 |
| 13.2.1.529 LUTIndex | 240 |
| 13.2.1.530 LUTSelector | 240 |
| 13.2.1.531 LUTValue | 240 |
| 13.2.1.532 LUTValueAll | 240 |
| 13.2.1.533 MaxDeviceResetTime | 241 |
| 13.2.1.534 OffsetX | 241 |
| 13.2.1.535 OffsetY | 241 |
| 13.2.1.536 PacketResendRequestCount | 241 |
| 13.2.1.537 PayloadSize | 241 |
| 13.2.1.538 PixelColorFilter | 241 |
| 13.2.1.539 PixelDynamicRangeMax | 241 |
| 13.2.1.540 PixelDynamicRangeMin | 241 |
| 13.2.1.541 PixelFormat | 242 |

| | |
|--|-----|
| 13.2.1.542 PixelFormatInfolD | 242 |
| 13.2.1.543 PixelFormatInfoSelector | 242 |
| 13.2.1.544 PixelSize | 242 |
| 13.2.1.545 PowerSupplyCurrent | 242 |
| 13.2.1.546 PowerSupplyVoltage | 242 |
| 13.2.1.547 RegionDestination | 242 |
| 13.2.1.548 RegionMode | 242 |
| 13.2.1.549 RegionSelector | 243 |
| 13.2.1.550 ReverseX | 243 |
| 13.2.1.551 ReverseY | 243 |
| 13.2.1.552 RgbTransformLightSource | 243 |
| 13.2.1.553 Saturation | 243 |
| 13.2.1.554 SaturationEnable | 243 |
| 13.2.1.555 Scan3dAxisMax | 243 |
| 13.2.1.556 Scan3dAxisMin | 243 |
| 13.2.1.557 Scan3dCoordinateOffset | 244 |
| 13.2.1.558 Scan3dCoordinateReferenceSelector | 244 |
| 13.2.1.559 Scan3dCoordinateReferenceValue | 244 |
| 13.2.1.560 Scan3dCoordinateScale | 244 |
| 13.2.1.561 Scan3dCoordinateSelector | 244 |
| 13.2.1.562 Scan3dCoordinateSystem | 244 |
| 13.2.1.563 Scan3dCoordinateSystemReference | 244 |
| 13.2.1.564 Scan3dCoordinateTransformSelector | 244 |
| 13.2.1.565 Scan3dDistanceUnit | 245 |
| 13.2.1.566 Scan3dInvalidDataFlag | 245 |
| 13.2.1.567 Scan3dInvalidDataValue | 245 |
| 13.2.1.568 Scan3dOutputMode | 245 |
| 13.2.1.569 Scan3dTransformValue | 245 |
| 13.2.1.570 SensorDescription | 245 |
| 13.2.1.571 SensorDigitizationTaps | 245 |
| 13.2.1.572 SensorHeight | 245 |
| 13.2.1.573 SensorShutterMode | 246 |
| 13.2.1.574 SensorTaps | 246 |
| 13.2.1.575 SensorWidth | 246 |
| 13.2.1.576 SequencerConfigurationMode | 246 |
| 13.2.1.577 SequencerConfigurationValid | 246 |
| 13.2.1.578 SequencerFeatureEnable | 246 |
| 13.2.1.579 SequencerMode | 246 |
| 13.2.1.580 SequencerPathSelector | 246 |
| 13.2.1.581 SequencerSetActive | 247 |
| 13.2.1.582 SequencerSetLoad | 247 |
| 13.2.1.583 SequencerSetNext | 247 |

| | |
|---|-----|
| 13.2.1.584 SequencerSetSave | 247 |
| 13.2.1.585 SequencerSetSelector | 247 |
| 13.2.1.586 SequencerSetStart | 247 |
| 13.2.1.587 SequencerSetValid | 247 |
| 13.2.1.588 SequencerTriggerActivation | 247 |
| 13.2.1.589 SequencerTriggerSource | 248 |
| 13.2.1.590 SerialPortBaudRate | 248 |
| 13.2.1.591 SerialPortDataBits | 248 |
| 13.2.1.592 SerialPortParity | 248 |
| 13.2.1.593 SerialPortSelector | 248 |
| 13.2.1.594 SerialPortSource | 248 |
| 13.2.1.595 SerialPortStopBits | 248 |
| 13.2.1.596 SerialReceiveFramingErrorCount | 248 |
| 13.2.1.597 SerialReceiveParityErrorCount | 249 |
| 13.2.1.598 SerialReceiveQueueClear | 249 |
| 13.2.1.599 SerialReceiveQueueCurrentCharacterCount | 249 |
| 13.2.1.600 SerialReceiveQueueMaxCharacterCount | 249 |
| 13.2.1.601 SerialTransmitQueueCurrentCharacterCount | 249 |
| 13.2.1.602 SerialTransmitQueueMaxCharacterCount | 249 |
| 13.2.1.603 Sharpening | 249 |
| 13.2.1.604 SharpeningAuto | 249 |
| 13.2.1.605 SharpeningEnable | 250 |
| 13.2.1.606 SharpeningThreshold | 250 |
| 13.2.1.607 SoftwareSignalPulse | 250 |
| 13.2.1.608 SoftwareSignalSelector | 250 |
| 13.2.1.609 SourceCount | 250 |
| 13.2.1.610 SourceSelector | 250 |
| 13.2.1.611 Test0001 | 250 |
| 13.2.1.612 TestEventGenerate | 250 |
| 13.2.1.613 TestPattern | 251 |
| 13.2.1.614 TestPatternGeneratorSelector | 251 |
| 13.2.1.615 TestPendingAck | 251 |
| 13.2.1.616 TimerDelay | 251 |
| 13.2.1.617 TimerDuration | 251 |
| 13.2.1.618 TimerReset | 251 |
| 13.2.1.619 TimerSelector | 251 |
| 13.2.1.620 TimerStatus | 251 |
| 13.2.1.621 TimerTriggerActivation | 252 |
| 13.2.1.622 TimerTriggerSource | 252 |
| 13.2.1.623 TimerValue | 252 |
| 13.2.1.624 Timestamp | 252 |
| 13.2.1.625 TimestampLatch | 252 |

| | |
|---|-----|
| 13.2.1.626 TimestampLatchValue | 252 |
| 13.2.1.627 TimestampReset | 252 |
| 13.2.1.628 TLParamsLocked | 252 |
| 13.2.1.629 TransferAbort | 253 |
| 13.2.1.630 TransferBlockCount | 253 |
| 13.2.1.631 TransferBurstCount | 253 |
| 13.2.1.632 TransferComponentSelector | 253 |
| 13.2.1.633 TransferControlMode | 253 |
| 13.2.1.634 TransferOperationMode | 253 |
| 13.2.1.635 TransferPause | 253 |
| 13.2.1.636 TransferQueueCurrentBlockCount | 253 |
| 13.2.1.637 TransferQueueMaxBlockCount | 254 |
| 13.2.1.638 TransferQueueMode | 254 |
| 13.2.1.639 TransferQueueOverflowCount | 254 |
| 13.2.1.640 TransferResume | 254 |
| 13.2.1.641 TransferSelector | 254 |
| 13.2.1.642 TransferStart | 254 |
| 13.2.1.643 TransferStatus | 254 |
| 13.2.1.644 TransferStatusSelector | 254 |
| 13.2.1.645 TransferStop | 255 |
| 13.2.1.646 TransferStreamChannel | 255 |
| 13.2.1.647 TransferTriggerActivation | 255 |
| 13.2.1.648 TransferTriggerMode | 255 |
| 13.2.1.649 TransferTriggerSelector | 255 |
| 13.2.1.650 TransferTriggerSource | 255 |
| 13.2.1.651 TriggerActivation | 255 |
| 13.2.1.652 TriggerDelay | 255 |
| 13.2.1.653 TriggerDivider | 256 |
| 13.2.1.654 TriggerEventTest | 256 |
| 13.2.1.655 TriggerMode | 256 |
| 13.2.1.656 TriggerMultiplier | 256 |
| 13.2.1.657 TriggerOverlap | 256 |
| 13.2.1.658 TriggerSelector | 256 |
| 13.2.1.659 TriggerSoftware | 256 |
| 13.2.1.660 TriggerSource | 256 |
| 13.2.1.661 UserOutputSelector | 257 |
| 13.2.1.662 UserOutputValue | 257 |
| 13.2.1.663 UserOutputValueAll | 257 |
| 13.2.1.664 UserOutputValueAllMask | 257 |
| 13.2.1.665 UserSetDefault | 257 |
| 13.2.1.666 UserSetFeatureEnable | 257 |
| 13.2.1.667 UserSetLoad | 257 |

| | |
|--|-----|
| 13.2.1.668 UserSetSave | 257 |
| 13.2.1.669 UserSetSelector | 258 |
| 13.2.1.670 V3_3Enable | 258 |
| 13.2.1.671 WhiteClip | 258 |
| 13.2.1.672 WhiteClipSelector | 258 |
| 13.2.1.673 Width | 258 |
| 13.2.1.674 WidthMax | 258 |
| 13.3 quickSpinTLDevice Struct Reference | 259 |
| 13.3.1 Field Documentation | 259 |
| 13.3.1.1 DeviceAccessStatus | 260 |
| 13.3.1.2 DeviceBootloaderVersion | 260 |
| 13.3.1.3 DeviceCurrentSpeed | 260 |
| 13.3.1.4 DeviceDisplayName | 260 |
| 13.3.1.5 DeviceDriverVersion | 260 |
| 13.3.1.6 DeviceEndianessMechanism | 260 |
| 13.3.1.7 DeviceID | 260 |
| 13.3.1.8 DeviceInstanceId | 260 |
| 13.3.1.9 DevicesUpdater | 261 |
| 13.3.1.10 DeviceLinkSpeed | 261 |
| 13.3.1.11 DeviceLocation | 261 |
| 13.3.1.12 DeviceModelName | 261 |
| 13.3.1.13 DeviceMulticastMonitorMode | 261 |
| 13.3.1.14 DevicePortId | 261 |
| 13.3.1.15 DeviceReset | 261 |
| 13.3.1.16 DeviceSerialNumber | 261 |
| 13.3.1.17 DeviceType | 262 |
| 13.3.1.18 DeviceU3VProtocol | 262 |
| 13.3.1.19 DeviceUserID | 262 |
| 13.3.1.20 DeviceVendorName | 262 |
| 13.3.1.21 DeviceVersion | 262 |
| 13.3.1.22 GenICamXMLLocation | 262 |
| 13.3.1.23 GenICamXMLPath | 262 |
| 13.3.1.24 GevCCP | 262 |
| 13.3.1.25 GevDeviceAutoForceIP | 263 |
| 13.3.1.26 GevDeviceDiscoverMaximumPacketSize | 263 |
| 13.3.1.27 GevDeviceForceGateway | 263 |
| 13.3.1.28 GevDeviceForceIP | 263 |
| 13.3.1.29 GevDeviceForceIPAddress | 263 |
| 13.3.1.30 GevDeviceForceSubnetMask | 263 |
| 13.3.1.31 GevDeviceGateway | 263 |
| 13.3.1.32 GevDeviceIPAddress | 263 |
| 13.3.1.33 GevDevicesWrongSubnet | 264 |

| | | |
|-----------|---------------------------------------|-----|
| 13.3.1.34 | GevDeviceMACAddress | 264 |
| 13.3.1.35 | GevDeviceMaximumPacketSize | 264 |
| 13.3.1.36 | GevDeviceMaximumRetryCount | 264 |
| 13.3.1.37 | GevDeviceModelsBigEndian | 264 |
| 13.3.1.38 | GevDevicePort | 264 |
| 13.3.1.39 | GevDeviceReadAndWriteTimeout | 264 |
| 13.3.1.40 | GevDeviceSubnetMask | 264 |
| 13.3.1.41 | GevVersionMajor | 265 |
| 13.3.1.42 | GevVersionMinor | 265 |
| 13.3.1.43 | GUIXMLLocation | 265 |
| 13.3.1.44 | GUIXMLPath | 265 |
| 13.4 | quickSpinTLInterface Struct Reference | 265 |
| 13.4.1 | Field Documentation | 266 |
| 13.4.1.1 | ActionCommand | 266 |
| 13.4.1.2 | DeviceAccessStatus | 266 |
| 13.4.1.3 | DeviceCount | 267 |
| 13.4.1.4 | DeviceID | 267 |
| 13.4.1.5 | DeviceModelName | 267 |
| 13.4.1.6 | DeviceSelector | 267 |
| 13.4.1.7 | DeviceSerialNumber | 267 |
| 13.4.1.8 | DeviceUnlock | 267 |
| 13.4.1.9 | DeviceUpdateList | 267 |
| 13.4.1.10 | DeviceVendorName | 267 |
| 13.4.1.11 | FLIRFilterDriverStatus | 268 |
| 13.4.1.12 | GevActionAckRequired | 268 |
| 13.4.1.13 | GevActionDeviceKey | 268 |
| 13.4.1.14 | GevActionGroupKey | 268 |
| 13.4.1.15 | GevActionGroupMask | 268 |
| 13.4.1.16 | GevActionTime | 268 |
| 13.4.1.17 | GevDeviceAutoForceIP | 268 |
| 13.4.1.18 | GevDeviceDisableDiscovery | 268 |
| 13.4.1.19 | GevDeviceDiscoveryEnabled | 269 |
| 13.4.1.20 | GevDeviceEnableDiscovery | 269 |
| 13.4.1.21 | GevDeviceForceGateway | 269 |
| 13.4.1.22 | GevDeviceForceIP | 269 |
| 13.4.1.23 | GevDeviceForceIPAddress | 269 |
| 13.4.1.24 | GevDeviceForceSubnetMask | 269 |
| 13.4.1.25 | GevDeviceGateway | 269 |
| 13.4.1.26 | GevDeviceIPAddress | 269 |
| 13.4.1.27 | GevDeviceMACAddress | 270 |
| 13.4.1.28 | GevDeviceSubnetMask | 270 |
| 13.4.1.29 | GevInterfaceGateway | 270 |

| | | |
|-----------|--------------------------------------|-----|
| 13.4.1.30 | GevInterfaceGatewaySelector | 270 |
| 13.4.1.31 | GevInterfaceIsIPConflict | 270 |
| 13.4.1.32 | GevInterfaceMACAddress | 270 |
| 13.4.1.33 | GevInterfaceMTU | 270 |
| 13.4.1.34 | GevInterfaceReceiveLinkSpeed | 270 |
| 13.4.1.35 | GevInterfaceSubnetIPAddress | 271 |
| 13.4.1.36 | GevInterfaceSubnetMask | 271 |
| 13.4.1.37 | GevInterfaceSubnetSelector | 271 |
| 13.4.1.38 | GevInterfaceTransmitLinkSpeed | 271 |
| 13.4.1.39 | HostAdapterDriverVersion | 271 |
| 13.4.1.40 | HostAdapterName | 271 |
| 13.4.1.41 | HostAdapterVendor | 271 |
| 13.4.1.42 | IncompatibleDeviceCount | 271 |
| 13.4.1.43 | IncompatibleDeviceID | 272 |
| 13.4.1.44 | IncompatibleDeviceModelName | 272 |
| 13.4.1.45 | IncompatibleDeviceSelector | 272 |
| 13.4.1.46 | IncompatibleDeviceVendorName | 272 |
| 13.4.1.47 | IncompatibleGevDeviceIPAddress | 272 |
| 13.4.1.48 | IncompatibleGevDeviceMACAddress | 272 |
| 13.4.1.49 | IncompatibleGevDeviceSubnetMask | 272 |
| 13.4.1.50 | InterfaceDisplayName | 272 |
| 13.4.1.51 | InterfaceID | 273 |
| 13.4.1.52 | InterfaceType | 273 |
| 13.4.1.53 | POEStatus | 273 |
| 13.4.1.54 | TeledyneGigeVisionFilterDriverStatus | 273 |
| 13.5 | quickSpinTLStream Struct Reference | 273 |
| 13.5.1 | Field Documentation | 274 |
| 13.5.1.1 | StreamAnnounceBufferMinimum | 274 |
| 13.5.1.2 | StreamAnnouncedBufferCount | 274 |
| 13.5.1.3 | StreamBlocksProcessingTimeLast | 274 |
| 13.5.1.4 | StreamBlocksProcessingTimeMax | 275 |
| 13.5.1.5 | StreamBlocksProcessingTimeMin | 275 |
| 13.5.1.6 | StreamBlocksReceptionTimeLast | 275 |
| 13.5.1.7 | StreamBlocksReceptionTimeMax | 275 |
| 13.5.1.8 | StreamBlocksReceptionTimeMin | 275 |
| 13.5.1.9 | StreamBlockTransferSize | 275 |
| 13.5.1.10 | StreamBufferAlignment | 275 |
| 13.5.1.11 | StreamBufferCountManual | 275 |
| 13.5.1.12 | StreamBufferCountMax | 276 |
| 13.5.1.13 | StreamBufferCountMode | 276 |
| 13.5.1.14 | StreamBufferCountResult | 276 |
| 13.5.1.15 | StreamBufferHandlingMode | 276 |

| | |
|--|-----|
| 13.5.1.16 StreamChunkCountMaximum | 276 |
| 13.5.1.17 StreamCRCCheckEnable | 276 |
| 13.5.1.18 StreamDeliveredFrameCount | 276 |
| 13.5.1.19 StreamDroppedFrameCount | 276 |
| 13.5.1.20 StreamID | 277 |
| 13.5.1.21 StreamIncompleteFrameCount | 277 |
| 13.5.1.22 StreamInputBufferCount | 277 |
| 13.5.1.23 StreamIsGrabbing | 277 |
| 13.5.1.24 StreamLostFrameCount | 277 |
| 13.5.1.25 StreamMissedPacketCount | 277 |
| 13.5.1.26 StreamMode | 277 |
| 13.5.1.27 StreamOutputBufferCount | 277 |
| 13.5.1.28 StreamPacketResendEnable | 278 |
| 13.5.1.29 StreamPacketResendMaxRequests | 278 |
| 13.5.1.30 StreamPacketResendReceivedPacketCount | 278 |
| 13.5.1.31 StreamPacketResendRequestCount | 278 |
| 13.5.1.32 StreamPacketResendRequestedPacketCount | 278 |
| 13.5.1.33 StreamPacketResendRequestTimeoutCount | 278 |
| 13.5.1.34 StreamPacketResendTimeout | 278 |
| 13.5.1.35 StreamPacketsDuplicatedCount | 278 |
| 13.5.1.36 StreamPacketsNotYetAvailableCount | 279 |
| 13.5.1.37 StreamPacketsPerFrameCount | 279 |
| 13.5.1.38 StreamPacketsTemporarilyUnavailableCount | 279 |
| 13.5.1.39 StreamPacketsTimeoutCount | 279 |
| 13.5.1.40 StreamPacketsUnavailableCount | 279 |
| 13.5.1.41 StreamReceivedFrameCount | 279 |
| 13.5.1.42 StreamReceivedPacketCount | 279 |
| 13.5.1.43 StreamStartedFrameCount | 279 |
| 13.5.1.44 StreamType | 280 |
| 13.6 quickSpinTLSystem Struct Reference | 280 |
| 13.6.1 Field Documentation | 280 |
| 13.6.1.1 EnumerateGen2Cameras | 280 |
| 13.6.1.2 EnumerateGEVInterfaces | 281 |
| 13.6.1.3 EnumerateUSBInterfaces | 281 |
| 13.6.1.4 GenTLFNCVersionMajor | 281 |
| 13.6.1.5 GenTLFNCVersionMinor | 281 |
| 13.6.1.6 GenTLFNCVersionSubMinor | 281 |
| 13.6.1.7 GenTLVersionMajor | 281 |
| 13.6.1.8 GenTLVersionMinor | 281 |
| 13.6.1.9 GevAutoAssignIPEnable | 281 |
| 13.6.1.10 GevInterfaceDefaultGateway | 282 |
| 13.6.1.11 GevInterfaceDefaultIPAddress | 282 |

| | | |
|-----------|--------------------------------|-----|
| 13.6.1.12 | GevInterfaceDefaultSubnetMask | 282 |
| 13.6.1.13 | GevInterfaceMACAddress | 282 |
| 13.6.1.14 | GevVersionMajor | 282 |
| 13.6.1.15 | GevVersionMinor | 282 |
| 13.6.1.16 | InterfaceDisplayName | 282 |
| 13.6.1.17 | InterfaceID | 282 |
| 13.6.1.18 | InterfaceSelector | 283 |
| 13.6.1.19 | InterfaceUpdateList | 283 |
| 13.6.1.20 | TLDisplayName | 283 |
| 13.6.1.21 | TLFileName | 283 |
| 13.6.1.22 | TLID | 283 |
| 13.6.1.23 | TLModelName | 283 |
| 13.6.1.24 | TLPath | 283 |
| 13.6.1.25 | TLType | 283 |
| 13.6.1.26 | TLVendorName | 284 |
| 13.6.1.27 | TLVersion | 284 |
| 13.7 | spinAVIOption Struct Reference | 284 |
| 13.7.1 | Detailed Description | 284 |
| 13.7.2 | Field Documentation | 284 |
| 13.7.2.1 | frameRate | 284 |
| 13.7.2.2 | height | 285 |
| 13.7.2.3 | reserved | 285 |
| 13.7.2.4 | width | 285 |
| 13.8 | spinBMPOption Struct Reference | 285 |
| 13.8.1 | Detailed Description | 285 |
| 13.8.2 | Field Documentation | 285 |
| 13.8.2.1 | indexedColor_8bit | 286 |
| 13.8.2.2 | reserved | 286 |
| 13.9 | spinChunkData Struct Reference | 286 |
| 13.9.1 | Detailed Description | 287 |
| 13.9.2 | Field Documentation | 287 |
| 13.9.2.1 | m_blackLevel | 287 |
| 13.9.2.2 | m_compressionMode | 287 |
| 13.9.2.3 | m_compressionRatio | 287 |
| 13.9.2.4 | m_counterValue | 287 |
| 13.9.2.5 | m_cRC | 288 |
| 13.9.2.6 | m_encoderValue | 288 |
| 13.9.2.7 | m_exposureEndLineStatusAll | 288 |
| 13.9.2.8 | m_exposureTime | 288 |
| 13.9.2.9 | m_frameID | 288 |
| 13.9.2.10 | m_gain | 288 |
| 13.9.2.11 | m_height | 288 |

| | |
|--|-----|
| 13.9.2.12 m_image | 288 |
| 13.9.2.13 m_inferenceConfidence | 289 |
| 13.9.2.14 m_inferenceFrameId | 289 |
| 13.9.2.15 m_inferenceResult | 289 |
| 13.9.2.16 m_linePitch | 289 |
| 13.9.2.17 m_lineStatusAll | 289 |
| 13.9.2.18 m_offsetX | 289 |
| 13.9.2.19 m_offsetY | 289 |
| 13.9.2.20 m_partSelector | 289 |
| 13.9.2.21 m_pixelDynamicRangeMax | 290 |
| 13.9.2.22 m_pixelDynamicRangeMin | 290 |
| 13.9.2.23 m_scan3dAxisMax | 290 |
| 13.9.2.24 m_scan3dAxisMin | 290 |
| 13.9.2.25 m_scan3dCoordinateOffset | 290 |
| 13.9.2.26 m_scan3dCoordinateReferenceValue | 290 |
| 13.9.2.27 m_scan3dCoordinateScale | 290 |
| 13.9.2.28 m_scan3dInvalidDataValue | 290 |
| 13.9.2.29 m_scan3dTransformValue | 291 |
| 13.9.2.30 m_scanLineSelector | 291 |
| 13.9.2.31 m_sequencerSetActive | 291 |
| 13.9.2.32 m_serialDataLength | 291 |
| 13.9.2.33 m_streamChannelID | 291 |
| 13.9.2.34 m_timerValue | 291 |
| 13.9.2.35 m_timestamp | 291 |
| 13.9.2.36 m_timestampLatchValue | 291 |
| 13.9.2.37 m_transferBlockID | 292 |
| 13.9.2.38 m_transferQueueCurrentBlockCount | 292 |
| 13.9.2.39 m_width | 292 |
| 13.10 spinH264Option Struct Reference | 292 |
| 13.10.1 Detailed Description | 292 |
| 13.10.2 Field Documentation | 292 |
| 13.10.2.1 bitrate | 293 |
| 13.10.2.2 frameRate | 293 |
| 13.10.2.3 height | 293 |
| 13.10.2.4 reserved | 293 |
| 13.10.2.5 width | 293 |
| 13.11 spinJPEGOOption Struct Reference | 293 |
| 13.11.1 Detailed Description | 294 |
| 13.11.2 Field Documentation | 294 |
| 13.11.2.1 progressive | 294 |
| 13.11.2.2 quality | 294 |
| 13.11.2.3 reserved | 294 |

| | |
|---|-----|
| 13.12 spinJPG2Option Struct Reference | 295 |
| 13.12.1 Detailed Description | 295 |
| 13.12.2 Field Documentation | 295 |
| 13.12.2.1 quality | 295 |
| 13.12.2.2 reserved | 295 |
| 13.13 spinLibraryVersion Struct Reference | 295 |
| 13.13.1 Detailed Description | 296 |
| 13.13.2 Field Documentation | 296 |
| 13.13.2.1 build | 296 |
| 13.13.2.2 major | 296 |
| 13.13.2.3 minor | 296 |
| 13.13.2.4 type | 297 |
| 13.14 spinMJPEGOption Struct Reference | 297 |
| 13.14.1 Detailed Description | 297 |
| 13.14.2 Field Documentation | 297 |
| 13.14.2.1 frameRate | 297 |
| 13.14.2.2 height | 298 |
| 13.14.2.3 quality | 298 |
| 13.14.2.4 reserved | 298 |
| 13.14.2.5 width | 298 |
| 13.15 spinPGMOption Struct Reference | 298 |
| 13.15.1 Detailed Description | 298 |
| 13.15.2 Field Documentation | 299 |
| 13.15.2.1 binaryFile | 299 |
| 13.15.2.2 reserved | 299 |
| 13.16 spinPNGOption Struct Reference | 299 |
| 13.16.1 Detailed Description | 299 |
| 13.16.2 Field Documentation | 299 |
| 13.16.2.1 compressionLevel | 300 |
| 13.16.2.2 interlaced | 300 |
| 13.16.2.3 reserved | 300 |
| 13.17 spinPPMOption Struct Reference | 300 |
| 13.17.1 Detailed Description | 300 |
| 13.17.2 Field Documentation | 300 |
| 13.17.2.1 binaryFile | 301 |
| 13.17.2.2 reserved | 301 |
| 13.18 spinTIFFOption Struct Reference | 301 |
| 13.18.1 Detailed Description | 301 |
| 13.18.2 Field Documentation | 301 |
| 13.18.2.1 compression | 301 |
| 13.18.2.2 reserved | 301 |

| | |
|---|------------|
| 14 File Documentation | 303 |
| 14.1 doc/spindocs/C/GettingStarted.dox File Reference | 303 |
| 14.2 doc/spindocs/C/ProgrammerGuide.dox File Reference | 303 |
| 14.3 doc/spindocs/shared/Benefits.dox File Reference | 303 |
| 14.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference | 303 |
| 14.5 doc/spindocs/shared/GenICamGenTL.dox File Reference | 303 |
| 14.6 doc/spindocs/shared/Licensing.dox File Reference | 303 |
| 14.7 doc/spindocs/shared/Maintenance.dox File Reference | 303 |
| 14.8 doc/spindocs/shared/NetworkingBestPractices.dox File Reference | 303 |
| 14.9 include/spinc/CameraDefsC.h File Reference | 303 |
| 14.10 include/spinc/ChunkDataDefC.h File Reference | 336 |
| 14.11 include/spinc/QuickSpinC.h File Reference | 337 |
| 14.11.1 Function Documentation | 337 |
| 14.11.1.1 quickSpinInit() | 337 |
| 14.11.1.2 quickSpinInitEx() | 338 |
| 14.11.1.3 quickSpinTLDeviceInit() | 338 |
| 14.11.1.4 quickSpinTLInterfaceInit() | 338 |
| 14.11.1.5 quickSpinTLStreamInit() | 338 |
| 14.11.1.6 quickSpinTLSystemInit() | 338 |
| 14.12 include/spinc/QuickSpinDefsC.h File Reference | 339 |
| 14.12.1 Typedef Documentation | 339 |
| 14.12.1.1 quickSpinBooleanNode | 340 |
| 14.12.1.2 quickSpinCommandNode | 340 |
| 14.12.1.3 quickSpinEnumerationNode | 340 |
| 14.12.1.4 quickSpinFloatNode | 340 |
| 14.12.1.5 quickSpinIntegerNode | 340 |
| 14.12.1.6 quickSpinRegisterNode | 340 |
| 14.12.1.7 quickSpinStringNode | 340 |
| 14.13 include/spinc/SpinnakerC.h File Reference | 341 |
| 14.13.1 Function Documentation | 351 |
| 14.13.1.1 spinCameraBeginAcquisition() | 351 |
| 14.13.1.2 spinCameraDeInit() | 351 |
| 14.13.1.3 spinCameraDiscoverMaxPacketSize() | 352 |
| 14.13.1.4 spinCameraEndAcquisition() | 352 |
| 14.13.1.5 spinCameraForceIP() | 352 |
| 14.13.1.6 spinCameraGetAccessMode() | 353 |
| 14.13.1.7 spinCameraGetDeviceID() | 353 |
| 14.13.1.8 spinCameraGetGuiXml() | 354 |
| 14.13.1.9 spinCameraGetNextImage() | 354 |
| 14.13.1.10 spinCameraGetNextImageEx() | 355 |
| 14.13.1.11 spinCameraGetNextImageSync() | 355 |
| 14.13.1.12 spinCameraGetNodeMap() | 356 |

| | |
|--|-----|
| 14.13.1.13 spinCameraGetTLDeviceNodeMap() | 356 |
| 14.13.1.14 spinCameraGetTLStreamNodeMap() | 357 |
| 14.13.1.15 spinCameraInit() | 357 |
| 14.13.1.16 spinCamerasInitialized() | 358 |
| 14.13.1.17 spinCamerasValid() | 358 |
| 14.13.1.18 spinCameraListAppend() | 359 |
| 14.13.1.19 spinCameraListClear() | 359 |
| 14.13.1.20 spinCameraListCreateEmpty() | 360 |
| 14.13.1.21 spinCameraListDestroy() | 360 |
| 14.13.1.22 spinCameraListGet() | 360 |
| 14.13.1.23 spinCameraListGetBySerial() | 361 |
| 14.13.1.24 spinCameraListGetSize() | 361 |
| 14.13.1.25 spinCameraListRemove() | 362 |
| 14.13.1.26 spinCameraListRemoveBySerial() | 362 |
| 14.13.1.27 spinCameraReadPort() | 363 |
| 14.13.1.28 spinCameraRegisterDeviceEventHandler() | 363 |
| 14.13.1.29 spinCameraRegisterDeviceEventHandlerEx() | 363 |
| 14.13.1.30 spinCameraRegisterImageEventHandler() | 364 |
| 14.13.1.31 spinCameraRegisterImageEventHandlerEx() | 364 |
| 14.13.1.32 spinCameraRegisterImageListEventHandler() | 365 |
| 14.13.1.33 spinCameraRelease() | 366 |
| 14.13.1.34 spinCameraUnregisterDeviceEventHandler() | 366 |
| 14.13.1.35 spinCameraUnregisterImageEventHandler() | 366 |
| 14.13.1.36 spinCameraUnregisterImageListEventHandler() | 367 |
| 14.13.1.37 spinCameraWritePort() | 367 |
| 14.13.1.38 spinDeviceArrivalEventHandlerCreate() | 368 |
| 14.13.1.39 spinDeviceArrivalEventHandlerDestroy() | 368 |
| 14.13.1.40 spinDeviceEventGetId() | 369 |
| 14.13.1.41 spinDeviceEventGetName() | 369 |
| 14.13.1.42 spinDeviceEventGetPayloadData() | 370 |
| 14.13.1.43 spinDeviceEventGetPayloadDataSize() | 370 |
| 14.13.1.44 spinDeviceEventHandlerCreate() | 371 |
| 14.13.1.45 spinDeviceEventHandlerDestroy() | 371 |
| 14.13.1.46 spinDeviceRemovalEventHandlerCreate() | 372 |
| 14.13.1.47 spinDeviceRemovalEventHandlerDestroy() | 372 |
| 14.13.1.48 spinErrorGetLast() | 373 |
| 14.13.1.49 spinErrorGetLastBuildDate() | 373 |
| 14.13.1.50 spinErrorGetLastBuildTime() | 373 |
| 14.13.1.51 spinErrorGetLastFileName() | 374 |
| 14.13.1.52 spinErrorGetLastFullMessage() | 374 |
| 14.13.1.53 spinErrorGetLastFunctionName() | 375 |
| 14.13.1.54 spinErrorGetLastLineNumber() | 375 |

| | |
|---|-----|
| 14.13.1.55 spinErrorGetLastMessage() | 376 |
| 14.13.1.56 spinImageCalculateStatistics() | 376 |
| 14.13.1.57 spinImageCheckCRC() | 377 |
| 14.13.1.58 spinImageChunkDataGetFloatValue() | 377 |
| 14.13.1.59 spinImageChunkDataGetIntValue() | 377 |
| 14.13.1.60 spinImageCreate() | 378 |
| 14.13.1.61 spinImageCreateEmpty() | 378 |
| 14.13.1.62 spinImageCreateEx() | 378 |
| 14.13.1.63 spinImageCreateEx2() | 379 |
| 14.13.1.64 spinImageDeepCopy() | 380 |
| 14.13.1.65 spinImageDestroy() | 380 |
| 14.13.1.66 spinImageEventHandlerCreate() | 381 |
| 14.13.1.67 spinImageEventHandlerDestroy() | 381 |
| 14.13.1.68 spinImageGetBitsPerPixel() | 382 |
| 14.13.1.69 spinImageGetBufferSize() | 382 |
| 14.13.1.70 spinImageGetChunkLayoutID() | 383 |
| 14.13.1.71 spinImageGetColorProcessing() | 383 |
| 14.13.1.72 spinImageGetData() | 384 |
| 14.13.1.73 spinImageGetFrameID() | 384 |
| 14.13.1.74 spinImageGetHeight() | 385 |
| 14.13.1.75 spinImageGetID() | 385 |
| 14.13.1.76 spinImageGetOffsetX() | 386 |
| 14.13.1.77 spinImageGetOffsetY() | 386 |
| 14.13.1.78 spinImageGetPaddingX() | 387 |
| 14.13.1.79 spinImageGetPaddingY() | 387 |
| 14.13.1.80 spinImageGetPayloadType() | 388 |
| 14.13.1.81 spinImageGetPixelFormat() | 388 |
| 14.13.1.82 spinImageGetPixelFormatName() | 389 |
| 14.13.1.83 spinImageGetPrivateData() | 389 |
| 14.13.1.84 spinImageGetSize() | 390 |
| 14.13.1.85 spinImageGetStatus() | 390 |
| 14.13.1.86 spinImageGetStatusDescription() | 391 |
| 14.13.1.87 spinImageGetStride() | 391 |
| 14.13.1.88 spinImageGetTimeStamp() | 392 |
| 14.13.1.89 spinImageGetTLPayloadType() | 392 |
| 14.13.1.90 spinImageGetTLPixelFormat() | 393 |
| 14.13.1.91 spinImageGetTLPixelFormatNamespace() | 393 |
| 14.13.1.92 spinImageGetValidPayloadSize() | 394 |
| 14.13.1.93 spinImageGetWidth() | 394 |
| 14.13.1.94 spinImageHasCRC() | 395 |
| 14.13.1.95 spinImageIsIncomplete() | 395 |
| 14.13.1.96 spinImageListAppend() | 396 |

| | |
|--|-----|
| 14.13.1.97 spinImageListClear() | 396 |
| 14.13.1.98 spinImageListCreateEmpty() | 397 |
| 14.13.1.99 spinImageListDestroy() | 397 |
| 14.13.1.100 spinImageListEventHandlerCreate() | 397 |
| 14.13.1.101 spinImageListEventHandlerDestroy() | 398 |
| 14.13.1.102 spinImageListGet() | 398 |
| 14.13.1.103 spinImageListGetByPixelFormat() | 399 |
| 14.13.1.104 spinImageListGetSize() | 399 |
| 14.13.1.105 spinImageListLoad() | 400 |
| 14.13.1.106 spinImageListRelease() | 400 |
| 14.13.1.107 spinImageListRemove() | 401 |
| 14.13.1.108 spinImageListRemoveByPixelFormat() | 401 |
| 14.13.1.109 spinImageListSave() | 402 |
| 14.13.1.110 spinImageProcessorApplyGamma() | 402 |
| 14.13.1.111 spinImageProcessorConvert() | 403 |
| 14.13.1.112 spinImageProcessorConvertImageList() | 403 |
| 14.13.1.113 spinImageProcessorCreate() | 404 |
| 14.13.1.114 spinImageProcessorDestroy() | 405 |
| 14.13.1.115 spinImageProcessorGetColorProcessing() | 405 |
| 14.13.1.116 spinImageProcessorGetNumDecompressionThreads() | 406 |
| 14.13.1.117 spinImageProcessorSetColorProcessing() | 406 |
| 14.13.1.118 spinImageProcessorSetNumDecompressionThreads() | 406 |
| 14.13.1.119 spinImageRelease() | 407 |
| 14.13.1.120 spinImageReset() | 407 |
| 14.13.1.121 spinImageResetEx() | 408 |
| 14.13.1.122 spinImageSave() | 409 |
| 14.13.1.123 spinImageSaveBmp() | 409 |
| 14.13.1.124 spinImageSaveFromExt() | 410 |
| 14.13.1.125 spinImageSaveJpeg() | 410 |
| 14.13.1.126 spinImageSaveJpg2() | 411 |
| 14.13.1.127 spinImageSavePgm() | 411 |
| 14.13.1.128 spinImageSavePng() | 412 |
| 14.13.1.129 spinImageSavePpm() | 412 |
| 14.13.1.130 spinImageSaveTiff() | 413 |
| 14.13.1.131 spinImageStatisticsCreate() | 413 |
| 14.13.1.132 spinImageStatisticsDestroy() | 413 |
| 14.13.1.133 spinImageStatisticsDisableAll() | 414 |
| 14.13.1.134 spinImageStatisticsEnableAll() | 414 |
| 14.13.1.135 spinImageStatisticsEnableGreyOnly() | 415 |
| 14.13.1.136 spinImageStatisticsEnableHslOnly() | 415 |
| 14.13.1.137 spinImageStatisticsEnableRgbOnly() | 416 |
| 14.13.1.138 spinImageStatisticsGetAll() | 416 |

| | |
|--|-----|
| 14.13.1.139 spinImageStatisticsGetChannelStatus() | 417 |
| 14.13.1.140 spinImageStatisticsGetHistogram() | 417 |
| 14.13.1.141 spinImageStatisticsGetMean() | 418 |
| 14.13.1.142 spinImageStatisticsGetNumPixelValues() | 418 |
| 14.13.1.143 spinImageStatisticsGetPixelValueRange() | 419 |
| 14.13.1.144 spinImageStatisticsGetRange() | 419 |
| 14.13.1.145 spinImageStatisticsSetChannelStatus() | 420 |
| 14.13.1.146 spinInterfaceEventHandlerCreate() | 420 |
| 14.13.1.147 spinInterfaceEventHandlerDestroy() | 421 |
| 14.13.1.148 spinInterfaceGetCameras() | 421 |
| 14.13.1.149 spinInterfaceGetCamerasEx() | 422 |
| 14.13.1.150 spinInterfaceGetTLNodeMap() | 422 |
| 14.13.1.151 spinInterfaceIsInUse() | 423 |
| 14.13.1.152 spinInterfaceListClear() | 423 |
| 14.13.1.153 spinInterfaceListCreateEmpty() | 424 |
| 14.13.1.154 spinInterfaceListDestroy() | 424 |
| 14.13.1.155 spinInterfaceListGet() | 424 |
| 14.13.1.156 spinInterfaceListGetSize() | 425 |
| 14.13.1.157 spinInterfaceRegisterDeviceArrivalEventHandler() | 425 |
| 14.13.1.158 spinInterfaceRegisterDeviceRemovalEventHandler() | 426 |
| 14.13.1.159 spinInterfaceRegisterInterfaceEventHandler() | 426 |
| 14.13.1.160 spinInterfaceRelease() | 427 |
| 14.13.1.161 spinInterfaceSendActionCommand() | 427 |
| 14.13.1.162 spinInterfaceUnregisterDeviceArrivalEventHandler() | 428 |
| 14.13.1.163 spinInterfaceUnregisterDeviceRemovalEventHandler() | 428 |
| 14.13.1.164 spinInterfaceUnregisterInterfaceEventHandler() | 429 |
| 14.13.1.165 spinInterfaceUpdateCameras() | 429 |
| 14.13.1.166 spinLogDataGetCategoryName() | 430 |
| 14.13.1.167 spinLogDataGetLogMessage() | 430 |
| 14.13.1.168 spinLogDataGetNDC() | 431 |
| 14.13.1.169 spinLogDataGetPriority() | 431 |
| 14.13.1.170 spinLogDataGetPriorityName() | 432 |
| 14.13.1.171 spinLogDataGetThreadName() | 432 |
| 14.13.1.172 spinLogDataGetTimestamp() | 433 |
| 14.13.1.173 spinLogEventHandlerCreate() | 433 |
| 14.13.1.174 spinLogEventHandlerDestroy() | 434 |
| 14.13.1.175 SPINNAKERC_API_DEPRECATED() | 434 |
| 14.13.1.176 spinSystemGetCameras() | 435 |
| 14.13.1.177 spinSystemGetCamerasEx() | 436 |
| 14.13.1.178 spinSystemGetInstance() | 436 |
| 14.13.1.179 spinSystemGetInterfaces() | 437 |
| 14.13.1.180 spinSystemGetLibraryVersion() | 437 |

| | |
|---|-----|
| 14.13.1.181 spinSystemGetLoggingLevel() | 438 |
| 14.13.1.182 spinSystemGetTLNodeMap() | 438 |
| 14.13.1.183 spinSystemIsInUse() | 439 |
| 14.13.1.184 spinSystemRegisterDeviceArrivalEventHandler() | 439 |
| 14.13.1.185 spinSystemRegisterDeviceRemovalEventHandler() | 440 |
| 14.13.1.186 spinSystemRegisterInterfaceEventHandler() | 440 |
| 14.13.1.187 spinSystemRegisterLogEventHandler() | 441 |
| 14.13.1.188 spinSystemReleaseInstance() | 441 |
| 14.13.1.189 spinSystemSendActionCommand() | 441 |
| 14.13.1.190 spinSystemSetLoggingLevel() | 442 |
| 14.13.1.191 spinSystemUnregisterAllLogEventHandlers() | 443 |
| 14.13.1.192 spinSystemUnregisterDeviceArrivalEventHandler() | 443 |
| 14.13.1.193 spinSystemUnregisterDeviceRemovalEventHandler() | 444 |
| 14.13.1.194 spinSystemUnregisterInterfaceEventHandler() | 444 |
| 14.13.1.195 spinSystemUnregisterLogEventHandler() | 445 |
| 14.13.1.196 spinSystemUpdateCameras() | 445 |
| 14.13.1.197 spinSystemUpdateCamerasEx() | 446 |
| 14.13.2 Variable Documentation | 446 |
| 14.13.2.1 pblsStreaming | 446 |
| 14.14 include/spinc/SpinnakerDefsC.h File Reference | 447 |
| 14.14.1 Typedef Documentation | 452 |
| 14.14.1.1 bool8_t | 452 |
| 14.14.1.2 spinArrivalEventFunction | 452 |
| 14.14.1.3 spinCamera | 452 |
| 14.14.1.4 spinCameraList | 452 |
| 14.14.1.5 spinDeviceArrivalEventHandler | 452 |
| 14.14.1.6 spinDeviceEventData | 452 |
| 14.14.1.7 spinDeviceEventFunction | 453 |
| 14.14.1.8 spinDeviceEventHandler | 453 |
| 14.14.1.9 spinDeviceRemovalEventHandler | 453 |
| 14.14.1.10 spinImage | 453 |
| 14.14.1.11 spinImageEventFunction | 453 |
| 14.14.1.12 spinImageEventHandler | 453 |
| 14.14.1.13 spinImageList | 454 |
| 14.14.1.14 spinImageListEventFunction | 454 |
| 14.14.1.15 spinImageListEventHandler | 454 |
| 14.14.1.16 spinImageProcessor | 454 |
| 14.14.1.17 spinImageStatistics | 454 |
| 14.14.1.18 spinInterface | 454 |
| 14.14.1.19 spinInterfaceEventHandler | 455 |
| 14.14.1.20 spinInterfaceList | 455 |
| 14.14.1.21 spinLogEventData | 455 |

| | |
|---|-----|
| 14.14.1.22 spinLogEventFunction | 455 |
| 14.14.1.23 spinLogEventHandler | 455 |
| 14.14.1.24 spinRemovalEventFunction | 455 |
| 14.14.1.25 spinSystem | 456 |
| 14.14.1.26 spinVideo | 456 |
| 14.14.2 Enumeration Type Documentation | 456 |
| 14.14.2.1 spinActionCodeStatus | 456 |
| 14.14.2.2 spinColorProcessingAlgorithm | 456 |
| 14.14.2.3 spinError | 457 |
| 14.14.2.4 spinImageFileFormat | 458 |
| 14.14.2.5 spinImageStatus | 459 |
| 14.14.2.6 spinnakerLogLevel | 460 |
| 14.14.2.7 spinStatisticsChannel | 460 |
| 14.14.2.8 spinTIFFCompressionMethod | 460 |
| 14.14.2.9 spinTLPayloadType | 461 |
| 14.14.2.10 spinTLPixelFormatNamespace | 461 |
| 14.14.3 Variable Documentation | 462 |
| 14.14.3.1 False | 462 |
| 14.14.3.2 True | 462 |
| 14.15 include/spinc/SpinnakerGenApiC.h File Reference | 462 |
| 14.15.1 Function Documentation | 466 |
| 14.15.1.1 spinBooleanGetValue() | 466 |
| 14.15.1.2 spinBooleanSetValue() | 467 |
| 14.15.1.3 spinCategoryGetFeatureByIndex() | 467 |
| 14.15.1.4 spinCategoryGetNumFeatures() | 468 |
| 14.15.1.5 spinCategoryReleaseNode() | 468 |
| 14.15.1.6 spinCommandExecute() | 469 |
| 14.15.1.7 spinCommandIsDone() | 469 |
| 14.15.1.8 spinEnumerationEntryGetEnumValue() | 470 |
| 14.15.1.9 spinEnumerationEntryGetIntValue() | 470 |
| 14.15.1.10 spinEnumerationEntryGetSymbolic() | 471 |
| 14.15.1.11 spinEnumerationGetCurrentEntry() | 471 |
| 14.15.1.12 spinEnumerationGetEntryByIndex() | 472 |
| 14.15.1.13 spinEnumerationGetEntryByName() | 472 |
| 14.15.1.14 spinEnumerationGetNumEntries() | 473 |
| 14.15.1.15 spinEnumerationReleaseNode() | 473 |
| 14.15.1.16 spinEnumerationSetEnumValue() | 474 |
| 14.15.1.17 spinEnumerationSetIntValue() | 474 |
| 14.15.1.18 spinFloatGetMax() | 475 |
| 14.15.1.19 spinFloatGetMin() | 475 |
| 14.15.1.20 spinFloatGetRepresentation() | 476 |
| 14.15.1.21 spinFloatGetUnit() | 476 |

| | |
|---|-----|
| 14.15.1.22 spinFloatGetValue() | 477 |
| 14.15.1.23 spinFloatGetValueEx() | 477 |
| 14.15.1.24 spinFloatSetValue() | 478 |
| 14.15.1.25 spinFloatSetValueEx() | 478 |
| 14.15.1.26 spinIntegerGetInc() | 479 |
| 14.15.1.27 spinIntegerGetMax() | 479 |
| 14.15.1.28 spinIntegerGetMin() | 480 |
| 14.15.1.29 spinIntegerGetRepresentation() | 480 |
| 14.15.1.30 spinIntegerGetValue() | 481 |
| 14.15.1.31 spinIntegerGetValueEx() | 481 |
| 14.15.1.32 spinIntegerSetValue() | 482 |
| 14.15.1.33 spinIntegerSetValueEx() | 482 |
| 14.15.1.34 spinNodeDeregisterCallback() | 483 |
| 14.15.1.35 spinNodeFromString() | 483 |
| 14.15.1.36 spinNodeFromStringEx() | 484 |
| 14.15.1.37 spinNodeGetAccessMode() | 484 |
| 14.15.1.38 spinNodeGetCachingMode() | 485 |
| 14.15.1.39 spinNodeGetDescription() | 485 |
| 14.15.1.40 spinNodeGetDisplayName() | 486 |
| 14.15.1.41 spinNodeGetImposedAccessMode() | 486 |
| 14.15.1.42 spinNodeGetImposedVisibility() | 487 |
| 14.15.1.43 spinNodeGetName() | 487 |
| 14.15.1.44 spinNodeGetNameSpace() | 488 |
| 14.15.1.45 spinNodeGetPollingTime() | 488 |
| 14.15.1.46 spinNodeGetToolTip() | 489 |
| 14.15.1.47 spinNodeGetType() | 489 |
| 14.15.1.48 spinNodeGetVisibility() | 490 |
| 14.15.1.49 spinNodeInvalidateNode() | 490 |
| 14.15.1.50 spinNodeIsAvailable() | 491 |
| 14.15.1.51 spinNodeIsEqual() | 491 |
| 14.15.1.52 spinNodeIsImplemented() | 492 |
| 14.15.1.53 spinNodeIsReadable() | 492 |
| 14.15.1.54 spinNodeIsWritable() | 492 |
| 14.15.1.55 spinNodeMapGetNode() | 493 |
| 14.15.1.56 spinNodeMapGetNodeByIndex() | 493 |
| 14.15.1.57 spinNodeMapGetNumNodes() | 494 |
| 14.15.1.58 spinNodeMapPoll() | 494 |
| 14.15.1.59 spinNodeMapReleaseNode() | 495 |
| 14.15.1.60 spinNodeRegisterCallback() | 495 |
| 14.15.1.61 spinNodeToString() | 496 |
| 14.15.1.62 spinNodeToStringEx() | 496 |
| 14.15.1.63 spinRegisterGet() | 497 |

| | |
|---|-----|
| 14.15.1.64 spinRegisterGetAddress() | 497 |
| 14.15.1.65 spinRegisterGetEx() | 498 |
| 14.15.1.66 spinRegisterGetLength() | 498 |
| 14.15.1.67 spinRegisterSet() | 499 |
| 14.15.1.68 spinRegisterSetEx() | 499 |
| 14.15.1.69 spinRegisterSetReference() | 500 |
| 14.15.1.70 spinStringGetMaxLength() | 500 |
| 14.15.1.71 spinStringGetValue() | 501 |
| 14.15.1.72 spinStringGetValueEx() | 501 |
| 14.15.1.73 spinStringSetValue() | 502 |
| 14.15.1.74 spinStringSetValueEx() | 502 |
| 14.16 include/spinc/SpinnakerGenApiDefsC.h File Reference | 503 |
| 14.16.1 Typedef Documentation | 506 |
| 14.16.1.1 spinNodeCallbackFunction | 506 |
| 14.16.1.2 spinNodeCallbackHandle | 506 |
| 14.16.1.3 spinNodeHandle | 506 |
| 14.16.1.4 spinNodeMapHandle | 506 |
| 14.16.2 Enumeration Type Documentation | 506 |
| 14.16.2.1 spinAccessMode | 506 |
| 14.16.2.2 spinCachingMode | 507 |
| 14.16.2.3 spinDisplayNotation | 507 |
| 14.16.2.4 spinEndianess | 508 |
| 14.16.2.5 spinIncMode | 508 |
| 14.16.2.6 spinInputDirection | 508 |
| 14.16.2.7 spinInterfaceType | 509 |
| 14.16.2.8 spinLinkType | 509 |
| 14.16.2.9 spinNameSpace | 510 |
| 14.16.2.10 spinNodeType | 510 |
| 14.16.2.11 spinRepresentation | 511 |
| 14.16.2.12 spinSign | 511 |
| 14.16.2.13 spinSlope | 511 |
| 14.16.2.14 spinStandardNameSpace | 512 |
| 14.16.2.15 spinVisibility | 512 |
| 14.16.2.16 spinXMLValidation | 513 |
| 14.16.2.17 spinYesNo | 513 |
| 14.17 include/spinc/SpinnakerPlatformC.h File Reference | 514 |
| 14.17.1 Macro Definition Documentation | 514 |
| 14.17.1.1 SPINNAKERC_API | 514 |
| 14.18 include/spinc/SpinVideoC.h File Reference | 515 |
| 14.18.1 Function Documentation | 515 |
| 14.18.1.1 spinVideoAppend() | 515 |
| 14.18.1.2 spinVideoClose() | 516 |

| | |
|---|------------|
| 14.18.1.3 spinVideoOpenH264() | 516 |
| 14.18.1.4 spinVideoOpenMJPEG() | 516 |
| 14.18.1.5 spinVideoOpenUncompressed() | 516 |
| 14.18.1.6 spinVideoSetMaximumFileSize() | 516 |
| 14.19 include/spinc/TransportLayerDefsC.h File Reference | 517 |
| 14.20 include/spinc/TransportLayerDeviceC.h File Reference | 519 |
| 14.21 include/spinc/TransportLayerInterfaceC.h File Reference | 520 |
| 14.22 include/spinc/TransportLayerStreamC.h File Reference | 520 |
| 14.23 include/spinc/TransportLayerSystemC.h File Reference | 521 |
| Index | 523 |

Chapter 1

Getting Started

The Spinnaker application programming interface (API) is used to interface with Teledyne'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](#)
- [Drivers](#)
- [Networking Best Practices](#)

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/00-FlyCapture-Comparison.html>) for the latest version of this document

Chapter 5

Working with GenICam GenTL Devices

5.1 GenTL Overview

Spinnaker 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 Teledyne 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 Teledyne 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 Spinnaker GenTL producer, it needs to be properly registered and installed on the system. **The Spinnaker 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 Spinnaker GenTL Logging

Spinnaker 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:

```
# Spinnaker 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 Spinnaker 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 | <p>Copyright (c) 2001-2023 FLIR Systems, Inc. All Rights Reserved.</p> <p>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).</p> <p>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.</p> |
| GenICam | GenICam License |
| AdapterList | The Code Project Open License (CPOL) |
| 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 (CPOL) 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 Teledyne maintenance policy for Spinnaker Software Development Kit (SDK). Teledyne 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

Teledyne 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

Teledyne 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

Teledyne 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

Teledyne 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

Networking Best Practices

8.1 GenTL Overview

First, is using a subnet mask of 255.255.255.0 (or /24 in CIDR). This is a common practice in many network setups and is ideal for our use case for the reasons stated below:

Network Size: A subnet mask of 255.255.255.0 allows for 256 possible IP addresses. This is suitable for small to medium size networks where you need to accommodate less than 256 devices. Our customers are not likely to use more than this number, however if this is ever the case we could resort to subnetting.

IP Address Allocation: You should plan your IP address allocation carefully to ensure efficient use of the available addresses within the subnet.

Performance: Smaller subnets can reduce broadcast traffic and improve network performance, since we don't need a lot of space for 256 cameras we might as well take advantage of this.

Basically, using a subnet mask of 255.255.255.0 is an ideal size for the use case of SpinView.

8.2 GenTL Overview

Guidelines for Configuring Multi-NIC Systems: When working with a computer that features multiple network interfaces, you must pay closer attention to networking settings to avoid difficult-to-debug connectivity problems. Follow these guidelines to ensure the best operation of your multi-NIC system. This includes any PC or embedded controller running a desktop OS.

Rule 1: Be Careful About Automatic IP Assignment (via DHCP or link-local addressing) Most Operation Systems are configured by default to obtain TCP/IP settings (IP address, subnet mask, and default gateway) automatically using a Dynamic Host Configuration Protocol (DHCP) server. If no DHCP server is found, then it is common practice for OSs to assign an IP address in the 169.254.x.x range, which is referred to as a link-local IP address.

Make sure that you are familiar with the DHCP server(s) on your network, and what IP address ranges and subnets they make use of. Likewise, make sure that you know if any adapters in your computer use DHCP and no DHCP server is on the network, resulting in the use of a 169.254.x.x address. If you are not familiar with the automatic IP assignment details for your network, then there is a much higher chance that you will break Rule 2 or Rule 3 below.

If possible given your network configuration, it is best to manually specify IP settings for each NIC in your multi-NIC PC. However, if you are on a network with other DHCP clients, using DHCP may be unavoidable.

Rule 2: Avoid Assigning Multiple NICs in the Same Computer to the Same Subnet Using multiple NICs on the same subnet is the #1 cause of connectivity issues on multi-NIC systems. While some OSs may be able to gracefully handle the presence of multiple NICs on one subnet, others may mistakenly attempt to send packets out of the wrong interface. You can remedy this issue by statically assigning one NIC (and the other networked computers attached to that NIC) to the 192.168.x.x range with subnet mask 255.255.0.0, and another NIC on the 10.0.x.x range with subnet mask 255.255.0.0. Note that this is just one possible configuration, any combination of two or more different subnets will work. It is recommended that you use one of the designated private IP address ranges to avoid conflicts with public servers on the internet.

In practice, there is rarely a good reason to use multiple NICs on the same subnet. One advanced configuration that warrants this setup is using multiple adapters to increase bandwidth is called adapter teaming, however this configuration is beyond the scope of this paper.

When configuring a controller with multiple NICs, each NIC should communicate with a different subnet. Configuring two or more NICs on the same subnet may cause communication problems.

One of the most common scenarios resulting in multiple NICs being assigned to the same subnet is when both a wired and wireless interface are used to connect to the same network. Once again, while some OSs may handle this case gracefully, it is a best practice to either configure the wired and wireless networks to exist on different subnets or to disable one network interface when using the other.

For more details reference: <https://www.ni.com/en/support/documentation/supplemental/11/best-practices-for-multiple-nics.html>

To summarize, we don't want to do this because it causes address conflicts, incompatibilities, and unpredictable behaviours.

Chapter 9

Module Index

9.1 Modules

Here is a list of all modules:

| | |
|---|-----|
| Spinnaker C QuickSpin API | 141 |
| Transport Layer Enumerations | 151 |
| TLDevice Structures | 159 |
| TLInterface Structures | 159 |
| TLStream Structures | 160 |
| TlSystem Structures | 160 |
| QuickSpin Access | 142 |
| Spinnaker C API | 142 |
| Spinnaker C Definitions | 23 |
| Camera Enumerations | 24 |
| Chunk Data Structures | 141 |
| Error Handling | 142 |
| System Access | 143 |
| InterfaceList Access | 143 |
| CameraList Access | 143 |
| ImageList Access | 143 |
| Interface Access | 143 |
| Camera Access | 143 |
| Image Access | 144 |
| Image Processor Access | 144 |
| Event Access | 147 |
| ImageStatistics Access | 147 |
| Logging Event Data Access | 147 |
| Device Event Data Access | 147 |
| Chunk data access | 147 |
| Spinnaker C Handles | 148 |
| Spinnaker C Function Signatures | 148 |
| Spinnaker C Enumerations | 148 |
| Spinnaker C Structures | 148 |
| Spinnaker C GenICam API | 148 |
| Node Map Access | 148 |
| Node Access | 148 |
| IValue Access | 149 |
| String Access | 149 |
| Integer Access | 149 |

| | |
|--|-----|
| IFloat Access | 149 |
| IEnumeration Access | 149 |
| IEnumEntry Access | 150 |
| IBoolean Access | 150 |
| ICommand Access | 150 |
| ICategory Access | 150 |
| IRegister Access | 150 |
| Spinnaker C GenICam Handles | 150 |
| Spinnaker C GenICam Enumerations | 151 |
| SpinVideo Recording Access | 151 |

Chapter 10

Data Structure Index

10.1 Data Structures

Here are the data structures with brief descriptions:

| | |
|--|-----|
| actionCommandResult | |
| Action Command Result | 161 |
| quickSpin | 162 |
| quickSpinTLDevice | 259 |
| quickSpinTLInterface | 265 |
| quickSpinTLStream | 273 |
| quickSpinTLSystem | 280 |
| spinAVIOption | |
| Options for saving uncompressed videos | 284 |
| spinBMPOption | |
| Options for saving BMP images | 285 |
| spinChunkData | |
| The type of information that can be obtained from image chunk data | 286 |
| spinH264Option | |
| Options for saving H264 videos | 292 |
| spinJPEGOption | |
| Options for saving JPEG images | 293 |
| spinJPG2Option | |
| Options for saving JPEG 2000 images | 295 |
| spinLibraryVersion | |
| Provides easier access to the current version of Spinnaker | 295 |
| spinMJPEGOption | |
| Options for saving MJPG videos | 297 |
| spinPGMOption | |
| Options for saving PGM images | 298 |
| spinPNGOption | |
| Options for saving PNG images | 299 |
| spinPPMOption | |
| Options for saving PPM images | 300 |
| spinTIFFOption | |
| Options for saving TIFF images | 301 |

Chapter 11

File Index

11.1 File List

Here is a list of all files with brief descriptions:

| | |
|---|-----|
| include/spinc/ CameraDefsC.h | 303 |
| include/spinc/ ChunkDataDefC.h | 336 |
| include/spinc/ QuickSpinC.h | 337 |
| include/spinc/ QuickSpinDefsC.h | 339 |
| include/spinc/ SpinnakerC.h | 341 |
| include/spinc/ SpinnakerDefsC.h | 447 |
| include/spinc/ SpinnakerGenApiC.h | 462 |
| include/spinc/ SpinnakerGenApiDefsC.h | 503 |
| include/spinc/ SpinnakerPlatformC.h | 514 |
| include/spinc/ SpinVideoC.h | 515 |
| include/spinc/ TransportLayerDefsC.h | 517 |
| include/spinc/ TransportLayerDeviceC.h | 519 |
| include/spinc/ TransportLayerInterfaceC.h | 520 |
| include/spinc/ TransportLayerStreamC.h | 520 |
| include/spinc/ TransportLayerSystemC.h | 521 |

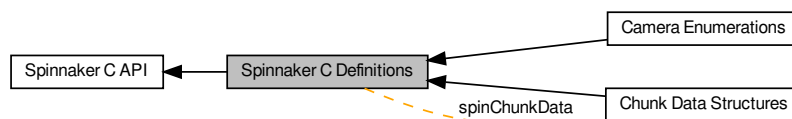
Chapter 12

Module Documentation

12.1 Spinnaker C Definitions

Definitions for Spinnaker C.

Collaboration diagram for Spinnaker C Definitions:



Modules

- [Camera Enumerations](#)
- [Chunk Data Structures](#)

Data Structures

- struct [spinChunkData](#)

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

12.1.1 Detailed Description

Definitions for Spinnaker C.

Definitions for Spinnaker C API.

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

12.2 Camera Enumerations

Collaboration diagram for Camera Enumerations:



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_DEVICE_TLTYPE }

• 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_DEVICETYPE }
• 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_DEVICESERIALPORTSELECTOR }
• enum spinDeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600 ,
DeviceSerialPortBaudRate_Baud19200 ,
DeviceSerialPortBaudRate_Baud38400 ,
DeviceSerialPortBaudRate_Baud57600 ,
DeviceSerialPortBaudRate_Baud115200 ,
DeviceSerialPortBaudRate_Baud230400 ,
DeviceSerialPortBaudRate_Baud460800 ,
DeviceSerialPortBaudRate_Baud921600 ,
NUM_DEVICESERIALPORTBAUDRATE }
• 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_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 spinCIConfigurationEnums {
    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 }

```

12.2.1 Detailed Description

12.2.2 Enumeration Type Documentation

12.2.2.1 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 | |

12.2.2.2 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 | |

12.2.2.3 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 | |

12.2.2.4 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 | |

12.2.2.5 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 | |

12.2.2.6 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 | |

12.2.2.7 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 | |

12.2.2.8 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

| | |
|--|--|
| <code>AutoExposureMeteringMode_Average</code> | |
| <code>AutoExposureMeteringMode_Spot</code> | |
| <code>AutoExposureMeteringMode_Partial</code> | |
| <code>AutoExposureMeteringMode_CenterWeighted</code> | |
| <code>AutoExposureMeteringMode_HistogramPeak</code> | |
| <code>NUM_AUTOEXPOSUREMETERINGMODE</code> | |

12.2.2.9 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

| | |
|---|--|
| <code>AutoExposureTargetGreyValueAuto_Off</code> | Target grey value is manually controlled |
| <code>AutoExposureTargetGreyValueAuto_Continuous</code> | Target grey value is constantly adapted by the device to maximize the dynamic range. |
| <code>NUM_AUTOEXPOSURETARGETGREYVALUEAUTO</code> | |

12.2.2.10 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 | |

12.2.2.11 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 | |

12.2.2.12 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 | |

12.2.2.13 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 | |

12.2.2.14 spinBinningSelectorEnums

enum `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 | |

12.2.2.15 spinBinningVerticalModeEnums

enum `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 | |

12.2.2.16 spinBlackLevelAutoBalanceEnums

enum `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 | |

12.2.2.17 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 | |

12.2.2.18 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 | |

12.2.2.19 spinChunkBlackLevelSelectorEnums

enum `spinChunkBlackLevelSelectorEnums`

< Selects which black level to retrieve

Enumerator

| | |
|-----------------------------|--|
| ChunkBlackLevelSelector_All | |
| NUM_CHUNKBLACKLEVELSELECTOR | |

12.2.2.20 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 | |

12.2.2.21 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 | |

12.2.2.22 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 | |

12.2.2.23 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 | |

12.2.2.24 spinChunkGainSelectorEnums

enum `spinChunkGainSelectorEnums`

< Selects which gain to retrieve

Enumerator

| | |
|-------------------------|--|
| ChunkGainSelector_All | |
| ChunkGainSelector_Red | |
| ChunkGainSelector_Green | |
| ChunkGainSelector_Blue | |
| NUM_CHUNKGAINSELECTOR | |

12.2.2.25 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. |
| ChunkImageComponent_Confidence | The image data is the confidence map component. |
| ChunkImageComponent_Scatter | The image data is the scatter component. |
| NUM_CHUNKIMAGECOMPONENT | |

12.2.2.26 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 | |

12.2.2.27 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 | |

12.2.2.28 spinChunkScan3dCoordinateReferenceSelectorEnums

enum `spinChunkScan3dCoordinateReferenceSelectorEnums`

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

Enumerator

| | |
|--|-------------------------|
| <code>ChunkScan3dCoordinateReferenceSelector_RotationX</code> | Rotation around X axis. |
| <code>ChunkScan3dCoordinateReferenceSelector_RotationY</code> | Rotation around Y axis. |
| <code>ChunkScan3dCoordinateReferenceSelector_RotationZ</code> | Rotation around Z axis. |
| <code>ChunkScan3dCoordinateReferenceSelector_TranslationX</code> | X axis translation. |
| <code>ChunkScan3dCoordinateReferenceSelector_TranslationY</code> | Y axis translation. |
| <code>ChunkScan3dCoordinateReferenceSelector_TranslationZ</code> | Z axis translation. |
| <code>NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR</code> | |

12.2.2.29 spinChunkScan3dCoordinateSelectorEnums

enum `spinChunkScan3dCoordinateSelectorEnums`

< Selects which Coordinate to retrieve data from.

Enumerator

| | |
|--|-----------------------------------|
| <code>ChunkScan3dCoordinateSelector_CoordinateA</code> | The first (X or Theta) coordinate |
| <code>ChunkScan3dCoordinateSelector_CoordinateB</code> | The second (Y or Phi) coordinate |
| <code>ChunkScan3dCoordinateSelector_CoordinateC</code> | The third (Z or Rho) coordinate. |
| <code>NUM_CHUNKSCAN3DCOORDINATESELECTOR</code> | |

12.2.2.30 spinChunkScan3dCoordinateSystemEnums

enum `spinChunkScan3dCoordinateSystemEnums`

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

Enumerator

| | |
|--|---|
| <code>ChunkScan3dCoordinateSystem_Cartesian</code> | Default value. 3-axis orthogonal, right-hand X-Y-Z. |
| <code>ChunkScan3dCoordinateSystem_Spherical</code> | A Theta-Phi-Rho coordinate system. |
| <code>ChunkScan3dCoordinateSystem_Cylindrical</code> | A Theta-Y-Rho coordinate system. |
| <code>NUM_CHUNKSCAN3DCOORDINATESYSTEM</code> | |

12.2.2.31 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_CHUNKSCAN3↔ DCOORDINATESYSTEMREFERENCE | |

12.2.2.32 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 | |

12.2.2.33 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 | |

12.2.2.34 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_PointCloud</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> | |

12.2.2.35 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 | |

12.2.2.36 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 | |

12.2.2.37 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 | |

12.2.2.38 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 | |

12.2.2.39 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 | |

12.2.2.40 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 | |

12.2.2.41 spinColorTransformationSelectorEnums

```
enum spinColorTransformationSelectorEnums
```

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

Enumerator

| | |
|--------------------------------------|--|
| ColorTransformationSelector_RGBtoRGB | |
| ColorTransformationSelector_RGBtoYUV | |
| NUM_COLORTRANSFORMATIONSELECTOR | |

12.2.2.42 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 | |

12.2.2.43 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 | |

12.2.2.44 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 | |

12.2.2.45 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 | |

12.2.2.46 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 | |

12.2.2.47 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 | |

12.2.2.48 spinCounterSelectorEnums

```
enum spinCounterSelectorEnums
```

< Selects which counter to configure

Enumerator

| | |
|--------------------------|--|
| CounterSelector_Counter0 | |
| CounterSelector_Counter1 | |
| NUM_COUNTERSELECTOR | |

12.2.2.49 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 | |

12.2.2.50 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 | |

12.2.2.51 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 | |

12.2.2.52 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> | |

12.2.2.53 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 | |

12.2.2.54 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 | |

12.2.2.55 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 | |

12.2.2.56 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 | |

12.2.2.57 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 | |

12.2.2.58 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 | |

12.2.2.59 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 | |

12.2.2.60 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 | |

12.2.2.61 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 | |

12.2.2.62 spinDeviceCharacterSetEnums

```
enum spinDeviceCharacterSetEnums
```

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

Enumerator

| | |
|--------------------------|--|
| DeviceCharacterSet_UTF8 | |
| DeviceCharacterSet_ASCII | |
| NUM_DEVICECHARACTERSET | |

12.2.2.63 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 | |

12.2.2.64 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 | |

12.2.2.65 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 | |

12.2.2.66 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 | |

12.2.2.67 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 | |

12.2.2.68 spinDevicePowerSupplySelectorEnums

enum `spinDevicePowerSupplySelectorEnums`

< Selects the power supply source to control or read.

Enumerator

| | |
|------------------------------------|--|
| DevicePowerSupplySelector_External | |
| NUM_DEVICEPOWERSUPPLYSELECTOR | |

12.2.2.69 spinDeviceRegistersEndiannessEnums

enum `spinDeviceRegistersEndiannessEnums`

< Endianness of the registers of the device.

Enumerator

| | |
|----------------------------------|--|
| DeviceRegistersEndianness_Little | |
| DeviceRegistersEndianness_Big | |
| NUM_DEVICEREGISTERSENDIANNES | |

12.2.2.70 spinDeviceScanTypeEnums

enum `spinDeviceScanTypeEnums`

< Scan type of the sensor of the device.

Enumerator

| | |
|-------------------------|--|
| DeviceScanType_Areascan | |
| NUM_DEVICEASCANTYPE | |

12.2.2.71 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 | |

12.2.2.72 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 | |

12.2.2.73 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_DEVICESTREAMCHANNELENDIANNESSE | |

12.2.2.74 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 | |

12.2.2.75 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 | |

12.2.2.76 spinDeviceTemperatureSelectorEnums

enum [spinDeviceTemperatureSelectorEnums](#)

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

Enumerator

| | |
|----------------------------------|--|
| DeviceTemperatureSelector_Sensor | |
| NUM_DEVICETEMPERATURESELECTOR | |

12.2.2.77 spinDeviceTLTypeEnums

enum [spinDeviceTLTypeEnums](#)

< Transport Layer type of the device.

Enumerator

| | |
|---------------------------|--|
| DeviceTLType_GigEVision | |
| DeviceTLType_CameraLink | |
| DeviceTLType_CameraLinkHS | |
| DeviceTLType_CoaXPress | |
| DeviceTLType_USB3Vision | |
| DeviceTLType_Custom | |
| NUM_DEVICETLTYPE | |

12.2.2.78 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 | |

12.2.2.79 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 | |

12.2.2.80 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 | |

12.2.2.81 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 | |

12.2.2.82 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 | |

12.2.2.83 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 | |

12.2.2.84 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 | |

12.2.2.85 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 | |

12.2.2.86 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 | |

12.2.2.87 spinEventNotificationEnums

```
enum spinEventNotificationEnums
```

< Enables/Disables the selected event.

Enumerator

| | |
|-----------------------|--|
| EventNotification_On | |
| EventNotification_Off | |
| NUM_EVENTNOTIFICATION | |

12.2.2.88 spinEventSelectorEnums

```
enum spinEventSelectorEnums
```

< Selects which Event to enable or disable.

Enumerator

| | |
|---------------------------------|--|
| EventSelector_Error | |
| EventSelector_ExposureEnd | |
| EventSelector_SerialPortReceive | |
| NUM_EVENTSELECTOR | |

12.2.2.89 spinExposureActiveModeEnums

```
enum spinExposureActiveModeEnums
```

< Control sensor active exposure mode.

Enumerator

| | |
|------------------------------|--|
| ExposureActiveMode_Line1 | |
| ExposureActiveMode_AnyPixels | |
| ExposureActiveMode_AllPixels | |
| NUM_EXPOSUREACTIVEMODE | |

12.2.2.90 spinExposureAutoEnums

enum `spinExposureAutoEnums`

< Sets the automatic exposure mode

Enumerator

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

12.2.2.91 spinExposureModeEnums

enum `spinExposureModeEnums`

< Sets the operation mode of the Exposure.

Enumerator

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

12.2.2.92 spinExposureTimeModeEnums

enum `spinExposureTimeModeEnums`

< Sets the configuration mode of the ExposureTime feature.

Enumerator

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

12.2.2.93 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 | |

12.2.2.94 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 | |

12.2.2.95 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 | |

12.2.2.96 spinFileOperationStatusEnums

enum `spinFileOperationStatusEnums`

< Represents the file operation execution status.

Enumerator

| | |
|------------------------------|--|
| FileOperationStatus_Success | File Operation was sucessful. |
| FileOperationStatus_Failure | File Operation failed. |
| FileOperationStatus_Overflow | An overflow occurred while executing the File Operation. |
| NUM_FILEOPERATIONSTATUS | |

12.2.2.97 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 | |

12.2.2.98 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 | |

12.2.2.99 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 | |

12.2.2.100 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 | |

12.2.2.101 spinGevCCPEnums

enum `spinGevCCPEnums`

< Controls the device access privilege of an application.

Enumerator

| | |
|------------------------|--|
| GevCCP_OpenAccess | |
| GevCCP_ExclusiveAccess | |
| GevCCP_ControlAccess | |
| NUM_GEVCCP | |

12.2.2.102 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 | |

12.2.2.103 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 | |

12.2.2.104 spinGevGVSPExtendedIDModeEnums

```
enum spinGevGVSPExtendedIDModeEnums
```

< Enables the extended IDs mode.

Enumerator

| | |
|---------------------------|-----|
| GevGVSPExtendedIDMode_Off | Off |
| GevGVSPExtendedIDMode_On | On |
| NUM_GEVGVSPEXTENDEDIDMODE | |

12.2.2.105 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 | |

12.2.2.106 spinGevIEEE1588ModeEnums

enum [spinGevIEEE1588ModeEnums](#)

< Provides the mode of the IEEE 1588 clock.

Enumerator

| | |
|---------------------------|------------|
| GevIEEE1588Mode_Auto | Automatic |
| GevIEEE1588Mode_SlaveOnly | Slave Only |
| NUM_GEVIEEE1588MODE | |

12.2.2.107 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 | |

12.2.2.108 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 | |

12.2.2.109 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 | |

12.2.2.110 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 | |

12.2.2.111 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 | |

12.2.2.112 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_↔ IMAGECOMPRESSIONJPEGFORMATOPTION | |

12.2.2.113 spinImageCompressionModeEnums

enum `spinImageCompressionModeEnums`

<

Enumerator

| | |
|-------------------------------|--|
| ImageCompressionMode_Off | |
| ImageCompressionMode_Lossless | |
| NUM_IMAGECOMPRESSIONMODE | |

12.2.2.114 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 | |

12.2.2.115 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 | |

12.2.2.116 spinLineInputFilterSelectorEnums

enum [spinLineInputFilterSelectorEnums](#)

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

Enumerator

| | |
|----------------------------------|--|
| LineInputFilterSelector_Deglitch | |
| LineInputFilterSelector_Debounce | |
| NUM_LINEINPUTFILTERSELECTOR | |

12.2.2.117 spinLineModeEnums

enum `spinLineModeEnums`

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

Enumerator

| | |
|-----------------|--|
| LineMode_Input | |
| LineMode_Output | |
| NUM_LINEMODE | |

12.2.2.118 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 | |

12.2.2.119 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 | |

12.2.2.120 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 | |

12.2.2.121 spinLogicBlockLUTInputSelectorEnums

```
enum spinLogicBlockLUTInputSelectorEnums
```

< Controls which LogicBlockLUT Input Source & Activation to access.

Enumerator

| | |
|-----------------------------------|--|
| LogicBlockLUTInputSelector_Input0 | |
| LogicBlockLUTInputSelector_Input1 | |
| LogicBlockLUTInputSelector_Input2 | |
| LogicBlockLUTInputSelector_Input3 | |
| NUM_LOGICBLOCKLUTINPUTSELECTOR | |

12.2.2.122 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 | |

12.2.2.123 spinLogicBlockLUTSelectorEnums

```
enum spinLogicBlockLUTSelectorEnums
```

< Selects which LogicBlock LUT to configure

Enumerator

| | |
|------------------------------|--|
| LogicBlockLUTSelector_Value | |
| LogicBlockLUTSelector_Enable | |
| NUM_LOGICBLOCKLUTSELECTOR | |

12.2.2.124 spinLogicBlockSelectorEnums

enum `spinLogicBlockSelectorEnums`

< Selects which LogicBlock to configure

Enumerator

| | |
|--------------------------------|--|
| LogicBlockSelector_LogicBlock0 | |
| LogicBlockSelector_LogicBlock1 | |
| NUM_LOGICBLOCKSELECTOR | |

12.2.2.125 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 | |

12.2.2.126 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 | |

12.2.2.127 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 | |

12.2.2.128 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_Cb↔ YYCrYY | 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_Cb↔ YCrY | 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_Cb↔ YCrY | 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_Cb↔ YYCrYY | 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_Cb↔ YCrY | 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_Cb↔YCrY | 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_UYYVYY | YUV 4:1:1 8-bit |
| PixelFormatInfoSelector_YUV422_8 | YUV 4:2:2 8-bit |
| PixelFormatInfoSelector_YUV422_8_UYVY | YUV 4:2:2 8-bit |
| PixelFormatInfoSelector_Polarized8 | Monochrome Polarized 8-bit |
| PixelFormatInfoSelector_Polarized10p | Monochrome Polarized 10-bit packed |
| PixelFormatInfoSelector_Polarized12p | Monochrome Polarized 12-bit packed |
| PixelFormatInfoSelector_Polarized16 | Monochrome Polarized 16-bit |
| PixelFormatInfoSelector_BayerRGPolarized8 | Polarized Bayer Red Green filter 8-bit |
| PixelFormatInfoSelector_BayerRGPolarized10p | Polarized Bayer Red Green filter 10-bit packed |
| PixelFormatInfoSelector_BayerRGPolarized12p | Polarized Bayer Red Green filter 12-bit packed |
| PixelFormatInfoSelector_BayerRGPolarized16 | Polarized Bayer Red Green filter 16-bit |
| 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 | |

12.2.2.129 spinPixelSizeEnums

```
enum spinPixelSizeEnums
```

< 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 | |

12.2.2.130 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 | |

12.2.2.131 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 | |

12.2.2.132 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 | |

12.2.2.133 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 | |

12.2.2.134 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 | |

12.2.2.135 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 | |

12.2.2.136 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 | |

12.2.2.137 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 | |

12.2.2.138 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 | |

12.2.2.139 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 | |

12.2.2.140 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 | |

12.2.2.141 spinSensorDigitizationTapsEnums

```
enum spinSensorDigitizationTapsEnums
```

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

Enumerator

| | |
|------------------------------|---------|
| SensorDigitizationTaps_One | 1 tap. |
| SensorDigitizationTaps_Two | 2 taps. |
| SensorDigitizationTaps_Three | 3 taps. |

Enumerator

| | |
|------------------------------|----------|
| SensorDigitizationTaps_Four | 4 taps. |
| SensorDigitizationTaps_Eight | 8 taps. |
| SensorDigitizationTaps_Ten | 10 taps. |
| NUM_SENSORDIGITIZATIONTAPS | |

12.2.2.142 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 | |

12.2.2.143 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 | |

12.2.2.144 spinSequencerConfigurationModeEnums

enum `spinSequencerConfigurationModeEnums`

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

Enumerator

| | |
|--------------------------------|--|
| SequencerConfigurationMode_Off | |
| SequencerConfigurationMode_On | |
| NUM_SEQUENCERCONFIGURATIONMODE | |

12.2.2.145 spinSequencerConfigurationValidEnums

enum `spinSequencerConfigurationValidEnums`

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

Enumerator

| | |
|---------------------------------|--|
| SequencerConfigurationValid_No | |
| SequencerConfigurationValid_Yes | |
| NUM_SEQUENCERCONFIGURATIONVALID | |

12.2.2.146 spinSequencerModeEnums

enum `spinSequencerModeEnums`

< Controls whether or not a sequencer is active.

Enumerator

| | |
|-------------------|--|
| SequencerMode_Off | |
| SequencerMode_On | |
| NUM_SEQUENCERMODE | |

12.2.2.147 spinSequencerSetValidEnums

enum `spinSequencerSetValidEnums`

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

Enumerator

| | |
|-----------------------|--|
| SequencerSetValid_No | |
| SequencerSetValid_Yes | |
| NUM_SEQUENCERSETVALID | |

12.2.2.148 spinSequencerTriggerActivationEnums

```
enum spinSequencerTriggerActivationEnums
```

< Specifies the activation mode of the sequencer trigger.

Enumerator

| | |
|--|--|
| SequencerTriggerActivation_RisingEdge | |
| SequencerTriggerActivation_FallingEdge | |
| SequencerTriggerActivation_AnyEdge | |
| SequencerTriggerActivation_LevelHigh | |
| SequencerTriggerActivation_LevelLow | |
| NUM_SEQUENCERTRIGGERACTIVATION | |

12.2.2.149 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 | |

12.2.2.150 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 | |

12.2.2.151 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 | |

12.2.2.152 spinSerialPortSelectorEnums

enum `spinSerialPortSelectorEnums`

< Selects which serial port of the device to control.

Enumerator

| | |
|--------------------------------|--|
| SerialPortSelector_SerialPort0 | |
| NUM_SERIALPORTSELECTOR | |

12.2.2.153 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 | |

12.2.2.154 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 | |

12.2.2.155 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 | |

12.2.2.156 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 | |

12.2.2.157 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 | |

12.2.2.158 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 | |

12.2.2.159 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 | |

12.2.2.160 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 | |

12.2.2.161 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 | |

12.2.2.162 spinTimerTriggerSourceEnums

enum `spinTimerTriggerSourceEnums`

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

Enumerator

| | |
|--|---|
| <code>TimerTriggerSource_Off</code> | Disables the Timer trigger. |
| <code>TimerTriggerSource_AcquisitionTrigger</code> | Starts with the reception of the Acquisition Trigger. |
| <code>TimerTriggerSource_AcquisitionStart</code> | Starts with the reception of the Acquisition Start. |
| <code>TimerTriggerSource_AcquisitionEnd</code> | Starts with the reception of the Acquisition End. |
| <code>TimerTriggerSource_FrameTrigger</code> | Starts with the reception of the Frame Start Trigger. |
| <code>TimerTriggerSource_FrameStart</code> | Starts with the reception of the Frame Start. |
| <code>TimerTriggerSource_FrameEnd</code> | Starts with the reception of the Frame End. |
| <code>TimerTriggerSource_FrameBurstStart</code> | Starts with the reception of the Frame Burst Start. |
| <code>TimerTriggerSource_FrameBurstEnd</code> | Starts with the reception of the Frame Burst End. |
| <code>TimerTriggerSource_LineTrigger</code> | Starts with the reception of the Line Start Trigger. |
| <code>TimerTriggerSource_LineStart</code> | Starts with the reception of the Line Start. |
| <code>TimerTriggerSource_LineEnd</code> | Starts with the reception of the Line End. |
| <code>TimerTriggerSource_ExposureStart</code> | Starts with the reception of the Exposure Start. |
| <code>TimerTriggerSource_ExposureEnd</code> | Starts with the reception of the Exposure End. |
| <code>TimerTriggerSource_Line0</code> | Starts when the specified <code>TimerTriggerActivation</code> condition is met on the chosen I/O Line. |
| <code>TimerTriggerSource_Line1</code> | Starts when the specified <code>TimerTriggerActivation</code> condition is met on the chosen I/O Line. |
| <code>TimerTriggerSource_Line2</code> | Starts when the specified <code>TimerTriggerActivation</code> condition is met on the chosen I/O Line. |
| <code>TimerTriggerSource_UserOutput0</code> | Specifies which User Output bit signal to use as internal source for the trigger. |
| <code>TimerTriggerSource_UserOutput1</code> | Specifies which User Output bit signal to use as internal source for the trigger. |
| <code>TimerTriggerSource_UserOutput2</code> | Specifies which User Output bit signal to use as internal source for the trigger. |
| <code>TimerTriggerSource_Counter0Start</code> | Starts with the reception of the Counter Start. |
| <code>TimerTriggerSource_Counter1Start</code> | Starts with the reception of the Counter Start. |
| <code>TimerTriggerSource_Counter2Start</code> | Starts with the reception of the Counter Start. |
| <code>TimerTriggerSource_Counter0End</code> | Starts with the reception of the Counter End. |
| <code>TimerTriggerSource_Counter1End</code> | Starts with the reception of the Counter End. |
| <code>TimerTriggerSource_Counter2End</code> | Starts with the reception of the Counter End. |
| <code>TimerTriggerSource_Timer0Start</code> | Starts with the reception of the Timer Start. |
| <code>TimerTriggerSource_Timer1Start</code> | Starts with the reception of the Timer Start. |
| <code>TimerTriggerSource_Timer2Start</code> | Starts with the reception of the Timer Start. |
| <code>TimerTriggerSource_Timer0End</code> | Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer. |
| <code>TimerTriggerSource_Timer1End</code> | Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer. |
| <code>TimerTriggerSource_Timer2End</code> | 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 | |

12.2.2.163 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 | |

12.2.2.164 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 | |

12.2.2.165 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 | |

12.2.2.166 spinTransferQueueModeEnums

enum `spinTransferQueueModeEnums`

< Specifies the operation mode of the transfer queue.

Enumerator

| | |
|-----------------------------------|--|
| TransferQueueMode_FirstInFirstOut | Blocks first In are transferred Out first. |
| NUM_TRANSFERQUEUEMODE | |

12.2.2.167 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 | |

12.2.2.168 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 | |

12.2.2.169 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 | |

12.2.2.170 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 | |

12.2.2.171 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 | |

12.2.2.172 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 | |

12.2.2.173 spinTriggerActivationEnums

```
enum spinTriggerActivationEnums
```

< Specifies the activation mode of the trigger.

Enumerator

| | |
|-------------------------------|--|
| TriggerActivation_LevelLow | |
| TriggerActivation_LevelHigh | |
| TriggerActivation_FallingEdge | |
| TriggerActivation_RisingEdge | |
| TriggerActivation_AnyEdge | |
| NUM_TRIGGERACTIVATION | |

12.2.2.174 spinTriggerModeEnums

enum `spinTriggerModeEnums`

< Controls whether or not trigger is active.

Enumerator

| | |
|-----------------|--|
| TriggerMode_Off | |
| TriggerMode_On | |
| NUM_TRIGGERMODE | |

12.2.2.175 spinTriggerOverlapEnums

enum `spinTriggerOverlapEnums`

< Specifies the overlap mode of the trigger.

Enumerator

| | |
|------------------------------|--|
| TriggerOverlap_Off | |
| TriggerOverlap_ReadOut | |
| TriggerOverlap_PreviousFrame | |
| NUM_TRIGGEROVERLAP | |

12.2.2.176 spinTriggerSelectorEnums

enum `spinTriggerSelectorEnums`

< Selects the type of trigger to configure.

Enumerator

| | |
|----------------------------------|--|
| TriggerSelector_AcquisitionStart | |
| TriggerSelector_FrameStart | |
| TriggerSelector_FrameBurstStart | |
| NUM_TRIGGERSELECTOR | |

12.2.2.177 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 | |

12.2.2.178 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 | |

12.2.2.179 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 | |

12.2.2.180 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 | |

12.2.2.181 spinWhiteClipSelectorEnums

enum `spinWhiteClipSelectorEnums`

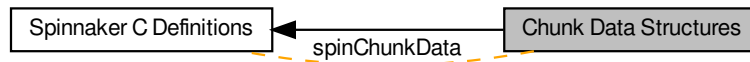
< Selects which White Clip to control.

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 | |

12.3 Chunk Data Structures

Collaboration diagram for Chunk Data Structures:



Data Structures

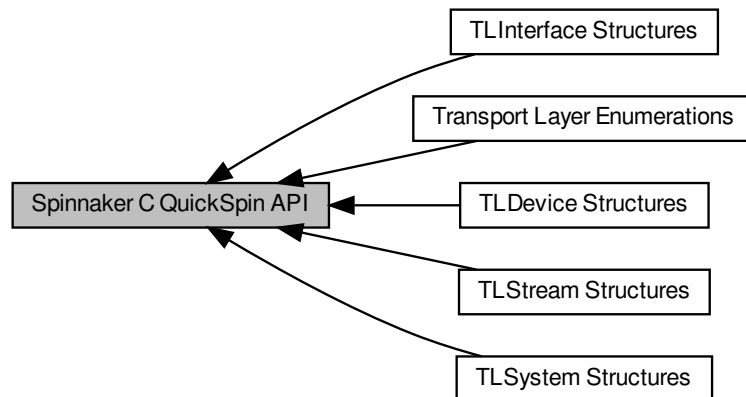
- struct [spinChunkData](#)

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

12.3.1 Detailed Description

12.4 Spinnaker C QuickSpin API

Collaboration diagram for Spinnaker C QuickSpin API:



Modules

- [Transport Layer Enumerations](#)
- [TLDevice Structures](#)
- [TLInterface Structures](#)
- [TLStream Structures](#)
- [TLSystem Structures](#)

12.4.1 Detailed Description

12.5 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.

12.6 Spinnaker C API

SpinnakerPlatform C Include.

Collaboration diagram for Spinnaker C API:



Modules

- [Spinnaker C Definitions](#)
Definitions for Spinnaker C.

12.6.1 Detailed Description

SpinnakerPlatform C Include.

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

Spinnaker C Definition Includes

12.7 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.

12.8 System Access

The functions in this section provide access to information, objects, and functionality of the system object.

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.

12.9 InterfaceList Access

The functions in this section provide access to information, objects, and functionality of interface lists.

The functions in this section provide access to information, objects, and functionality of interface lists.

This includes updating, size and interface retrieval, and clearance.

12.10 CameraList Access

The functions in this section provide access to information, objects, and functionality of camera lists.

The functions in this section provide access to information, objects, and functionality of camera lists.

This includes updating, size and camera retrieval, and clearance.

12.11 ImageList Access

The functions in this section provide access to information, objects, and functionality of image lists.

The functions in this section provide access to information, objects, and functionality of image lists.

This includes updating, size and image retrieval, and clearance.

12.12 Interface Access

The functions in this section provide access to information, objects, and functionality of interfaces.

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.

12.13 Camera Access

The functions in this section provide access to information, objects, and functionality of cameras.

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.

12.14 Image Access

The functions in this section provide access to information and functionality of images.

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.

12.15 Image Processor Access

The functions in this section provide access to information and functionality of image processor.

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

12.16 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.

12.17 ImageStatistics Access

The functions in this section provide access to information and functionality related to image statistics.

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.

12.18 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.

12.19 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.

12.20 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.

12.21 Spinnaker C Handles

Spinnaker C handle definitions.

Spinnaker C handle definitions.

12.22 Spinnaker C Function Signatures

Spinnaker C function signature definitions.

Spinnaker C function signature definitions.

12.23 Spinnaker C Enumerations

Spinnaker C enumeration definitions.

Spinnaker C enumeration definitions.

12.24 Spinnaker C Structures

Spinnaker C structure definitions.

Spinnaker C structure definitions.

12.25 Spinnaker C GenICam API

12.26 Node Map Access

The functions in this section provide access to information, objects, and functionality related to nodemaps.

The functions in this section provide access to information, objects, and functionality related to nodemaps.

This includes nodes, node counts, and polling.

12.27 Node Access

The functions in this section provide access to information and objects retrieved from nodes.

The functions in this section provide access to information and objects retrieved from nodes.

This includes node properties and callback registration.

12.28 IValue Access

The functions in this section provide access to nodes as value nodes.

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.

12.29 String Access

The functions in this section provide access to string nodes using character pointers and arrays.

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.

12.30 Integer Access

The functions in this section provide access to integer nodes using the `int64_t` data type.

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.

12.31 IFloat Access

The functions in this section provide access to float nodes using `double` as the data type.

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.

12.32 IEnumeration Access

The functions in this section provide access to enum nodes.

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.

12.33 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.

12.34 IBoolean Access

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

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.

12.35 ICommand Access

The functions in this section all provide access to information and objects retrieved from nodes.

The functions in this section all provide access to information and objects retrieved from nodes.

This includes node properties and callbacks.

12.36 ICategory Access

The functions in this section all provide access to information and objects retrieved from nodes.

The functions in this section all provide access to information and objects retrieved from nodes.

This includes node properties and callbacks.

12.37 IRegister Access

The functions in this section provide access to register nodes.

The functions in this section provide access to register nodes.

This includes access to the node, its address and length, and reference.

12.38 Spinnaker C GenICam Handles

Handle definitions for Spinnaker C GenICam API.

Handle definitions for Spinnaker C GenICam API.

12.39 Spinnaker C GenICam Enumerations

Enumeration definitions for Spinnaker C GenICam API.

Enumeration definitions for Spinnaker C GenICam API.

12.40 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.

12.41 Transport Layer Enumerations

Collaboration diagram for Transport Layer Enumerations:



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_TeledyneGigeVision` ,
`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 `spinTLDeviceEndiannessMechanismEnums` {
`DeviceEndiannessMechanism_Legacy` ,
`DeviceEndiannessMechanism_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 `spinTLFLIRFilterDriverStatusEnums` {
`FLIRFilterDriverStatus_NotSupported` ,
`FLIRFilterDriverStatus_Disabled` ,

```

    FLIRFilterDriverStatus_Enabled ,
    NUMFLIRFILTERDRIVERSTATUS }
• enum spinTLTeledyneGigeVisionFilterDriverStatusEnums {
    TeledyneGigeVisionFilterDriverStatus_NotSupported ,
    TeledyneGigeVisionFilterDriverStatus_Disabled ,
    TeledyneGigeVisionFilterDriverStatus_Enabled ,
    NUMTELEDYNEGIGEVISIONFILTERDRIVERSTATUS }
• enum spinTLTLTypeEnums {
    TLType_GigEVision ,
    TLType_CameraLink ,
    TLType_CameraLinkHS ,
    TLType_CoaXPRESS ,
    TLType_USB3Vision ,
    TLType_Mixed ,
    TLType_Custom ,
    NUMTLTYPE }

```

12.41.1 Detailed Description

12.41.2 Enumeration Type Documentation

12.41.2.1 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 | |

12.41.2.2 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 | |

12.41.2.3 spinTLDeviceEndiannessMechanismEnums

enum [spinTLDeviceEndiannessMechanismEnums](#)

< 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 | |

12.41.2.4 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 | |

12.41.2.5 spinTLFLIRFilterDriverStatusEnums

enum [spinTLFLIRFilterDriverStatusEnums](#)

< Reports whether FLIR Light Weight Filter Driver is enabled, disabled, or not installed.

Enumerator

| | |
|-------------------------------------|---|
| FLIRFilterDriverStatus_NotSupported | Not Installed |
| FLIRFilterDriverStatus_Disabled | FLIR Light Weight Filter Driver is disabled across all interfaces |
| FLIRFilterDriverStatus_Enabled | FLIR Light Weight Filter Driver is enabled |
| NUMFLIRFILTERDRIVERSTATUS | |

12.41.2.6 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 | |

12.41.2.7 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 | |

12.41.2.8 spinTLGUIXMLLocationEnums

enum `spinTLGUIXMLLocationEnums`

< Sets the location to load GUI XML.

Enumerator

| | |
|-----------------------|----------------------|
| GUIXMLLocation_Device | Load XML from device |
| GUIXMLLocation_Host | Load XML from host |
| NUMGUIXMLLOCATION | |

12.41.2.9 spinTLInterfaceTypeEnums

enum `spinTLInterfaceTypeEnums`

< Transport layer type of the interface.

Enumerator

| | |
|----------------------------|------------------------|
| InterfaceType_GigEVision | GigE Vision |
| InterfaceType_CameraLink | Camera Link |
| InterfaceType_CameraLinkHS | Camera Link High Speed |
| InterfaceType_CoaXPress | CoaXPress |
| InterfaceType_USB3Vision | USB3 Vision |
| InterfaceType_Custom | Custom transport layer |
| NUMINTERFACETYPE | |

12.41.2.10 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 | |

12.41.2.11 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 | |

12.41.2.12 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 | |

12.41.2.13 spinTLStreamModeEnums

```
enum spinTLStreamModeEnums
```

< Stream mode of the device.

Enumerator

| | |
|-------------------------------|-----------------------------|
| StreamMode_Socket | Socket |
| StreamMode_LWF | Light Weight Filter Driver |
| StreamMode_TeledyneGigeVision | Teledyne Gige Vision Driver |
| NUMSTREAMMODE | |

12.41.2.14 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 | |

12.41.2.15 spinTLTeledyneGigeVisionFilterDriverStatusEnums

enum [spinTLTeledyneGigeVisionFilterDriverStatusEnums](#)

< Reports whether Teledyne Gige Vision Filter Driver is enabled, disabled, or not installed.

Enumerator

| | |
|---|--|
| TeledyneGigeVisionFilterDriverStatus_NotSupported | Not Installed |
| TeledyneGigeVisionFilterDriverStatus_Disabled | Teledyne Gige Vision Filter Driver is disabled across all interfaces |
| TeledyneGigeVisionFilterDriverStatus_Enabled | Teledyne Gige Vision Filter Driver is enabled |
| NUMTELEDYNEGIGEVISIONFILTERDRIVERSTATUS | |

12.41.2.16 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 | |

12.42 TLDevice Structures

Collaboration diagram for TLDevice Structures:



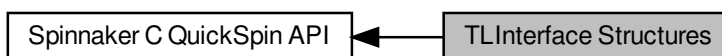
Data Structures

- struct [quickSpinTLDevice](#)

12.42.1 Detailed Description

12.43 TLInterface Structures

Collaboration diagram for TLInterface Structures:



Data Structures

- struct [quickSpinTLInterface](#)

12.43.1 Detailed Description

12.44 TLStream Structures

Collaboration diagram for TLStream Structures:



Data Structures

- struct [quickSpinTLStream](#)

12.44.1 Detailed Description

12.45 TLSystem Structures

Collaboration diagram for TLSystem Structures:



Data Structures

- struct [quickSpinTLSystem](#)

12.45.1 Detailed Description

Chapter 13

Data Structure Documentation

13.1 actionCommandResult Struct Reference

Action Command Result.

Data Fields

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

13.1.1 Detailed Description

Action Command Result.

13.1.2 Field Documentation

13.1.2.1 DeviceAddress

```
unsigned int DeviceAddress
```

13.1.2.2 Status

```
spinActionCommandStatus Status
```

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

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

13.2 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](#)

13.2.1 Field Documentation

13.2.1.1 AasRoiEnable

[quickSpinBooleanNode](#) AasRoiEnable

13.2.1.2 AasRoiHeight

[quickSpinIntegerNode](#) AasRoiHeight

13.2.1.3 AasRoiOffsetX

[quickSpinIntegerNode](#) AasRoiOffsetX

13.2.1.4 AasRoiOffsetY

[quickSpinIntegerNode](#) AasRoiOffsetY

13.2.1.5 AasRoiWidth

`quickSpinIntegerNode` AasRoiWidth

13.2.1.6 AcquisitionAbort

`quickSpinCommandNode` AcquisitionAbort

13.2.1.7 AcquisitionArm

`quickSpinCommandNode` AcquisitionArm

13.2.1.8 AcquisitionBurstFrameCount

`quickSpinIntegerNode` AcquisitionBurstFrameCount

13.2.1.9 AcquisitionFrameCount

`quickSpinIntegerNode` AcquisitionFrameCount

13.2.1.10 AcquisitionFrameRate

`quickSpinFloatNode` AcquisitionFrameRate

13.2.1.11 AcquisitionFrameRateEnable

`quickSpinBooleanNode` AcquisitionFrameRateEnable

13.2.1.12 AcquisitionLineRate

`quickSpinFloatNode` AcquisitionLineRate

13.2.1.13 AcquisitionMode

`quickSpinEnumerationNode` AcquisitionMode

13.2.1.14 AcquisitionResultingFrameRate

`quickSpinFloatNode` AcquisitionResultingFrameRate

13.2.1.15 AcquisitionStart

`quickSpinCommandNode` AcquisitionStart

13.2.1.16 AcquisitionStatus

`quickSpinBooleanNode` AcquisitionStatus

13.2.1.17 AcquisitionStatusSelector

`quickSpinEnumerationNode` AcquisitionStatusSelector

13.2.1.18 AcquisitionStop

`quickSpinCommandNode` AcquisitionStop

13.2.1.19 ActionDeviceKey

`quickSpinIntegerNode` ActionDeviceKey

13.2.1.20 ActionGroupKey

`quickSpinIntegerNode` ActionGroupKey

13.2.1.21 ActionGroupMask

`quickSpinIntegerNode` ActionGroupMask

13.2.1.22 ActionQueueSize

`quickSpinIntegerNode` ActionQueueSize

13.2.1.23 ActionSelector

`quickSpinIntegerNode` ActionSelector

13.2.1.24 ActionUnconditionalMode

`quickSpinEnumerationNode` ActionUnconditionalMode

13.2.1.25 AdaptiveCompressionEnable

`quickSpinBooleanNode` AdaptiveCompressionEnable

13.2.1.26 AdcBitDepth

`quickSpinEnumerationNode` AdcBitDepth

13.2.1.27 aPAUSEMACCtrlFramesReceived

`quickSpinIntegerNode` aPAUSEMACCtrlFramesReceived

13.2.1.28 aPAUSEMACCtrlFramesTransmitted

`quickSpinIntegerNode` aPAUSEMACCtrlFramesTransmitted

13.2.1.29 AutoAlgorithmSelector

`quickSpinEnumerationNode` AutoAlgorithmSelector

13.2.1.30 AutoExposureControlLoopDamping

`quickSpinFloatNode` AutoExposureControlLoopDamping

13.2.1.31 AutoExposureControlPriority

`quickSpinEnumerationNode` AutoExposureControlPriority

13.2.1.32 AutoExposureEVCompensation

`quickSpinFloatNode` AutoExposureEVCompensation

13.2.1.33 AutoExposureExposureTimeLowerLimit

`quickSpinFloatNode` AutoExposureExposureTimeLowerLimit

13.2.1.34 AutoExposureExposureTimeUpperLimit

`quickSpinFloatNode` AutoExposureExposureTimeUpperLimit

13.2.1.35 AutoExposureGainLowerLimit

`quickSpinFloatNode` AutoExposureGainLowerLimit

13.2.1.36 AutoExposureGainUpperLimit

`quickSpinFloatNode` AutoExposureGainUpperLimit

13.2.1.37 AutoExposureGreyValueLowerLimit

`quickSpinFloatNode` AutoExposureGreyValueLowerLimit

13.2.1.38 AutoExposureGreyValueUpperLimit

`quickSpinFloatNode` AutoExposureGreyValueUpperLimit

13.2.1.39 AutoExposureLightingMode

`quickSpinEnumerationNode` AutoExposureLightingMode

13.2.1.40 AutoExposureMeteringMode

`quickSpinEnumerationNode` AutoExposureMeteringMode

13.2.1.41 AutoExposureTargetGreyValue

`quickSpinFloatNode` AutoExposureTargetGreyValue

13.2.1.42 AutoExposureTargetGreyValueAuto

`quickSpinEnumerationNode` AutoExposureTargetGreyValueAuto

13.2.1.43 BalanceRatio

`quickSpinFloatNode` BalanceRatio

13.2.1.44 BalanceRatioSelector

`quickSpinEnumerationNode` BalanceRatioSelector

13.2.1.45 BalanceWhiteAuto

`quickSpinEnumerationNode` BalanceWhiteAuto

13.2.1.46 BalanceWhiteAutoDamping

`quickSpinFloatNode` BalanceWhiteAutoDamping

13.2.1.47 BalanceWhiteAutoLowerLimit

`quickSpinFloatNode` BalanceWhiteAutoLowerLimit

13.2.1.48 BalanceWhiteAutoProfile

`quickSpinEnumerationNode` BalanceWhiteAutoProfile

13.2.1.49 BalanceWhiteAutoUpperLimit

`quickSpinFloatNode` BalanceWhiteAutoUpperLimit

13.2.1.50 BinningHorizontal

`quickSpinIntegerNode` BinningHorizontal

13.2.1.51 BinningHorizontalMode

`quickSpinEnumerationNode` BinningHorizontalMode

13.2.1.52 BinningSelector

`quickSpinEnumerationNode` BinningSelector

13.2.1.53 BinningVertical

`quickSpinIntegerNode` BinningVertical

13.2.1.54 BinningVerticalMode

`quickSpinEnumerationNode` BinningVerticalMode

13.2.1.55 BlackLevel

`quickSpinFloatNode` BlackLevel

13.2.1.56 BlackLevelAuto

`quickSpinEnumerationNode` BlackLevelAuto

13.2.1.57 BlackLevelAutoBalance

`quickSpinEnumerationNode` BlackLevelAutoBalance

13.2.1.58 BlackLevelClampingEnable

`quickSpinBooleanNode` BlackLevelClampingEnable

13.2.1.59 BlackLevelRaw

`quickSpinIntegerNode` BlackLevelRaw

13.2.1.60 BlackLevelSelector

`quickSpinEnumerationNode` BlackLevelSelector

13.2.1.61 ChunkBlackLevel

`quickSpinFloatNode` `ChunkBlackLevel`

13.2.1.62 ChunkBlackLevelSelector

`quickSpinEnumerationNode` `ChunkBlackLevelSelector`

13.2.1.63 ChunkCompressionMode

`quickSpinIntegerNode` `ChunkCompressionMode`

13.2.1.64 ChunkCompressionRatio

`quickSpinFloatNode` `ChunkCompressionRatio`

13.2.1.65 ChunkCounterSelector

`quickSpinEnumerationNode` `ChunkCounterSelector`

13.2.1.66 ChunkCounterValue

`quickSpinIntegerNode` `ChunkCounterValue`

13.2.1.67 ChunkCRC

`quickSpinIntegerNode` `ChunkCRC`

13.2.1.68 ChunkEnable

`quickSpinBooleanNode` `ChunkEnable`

13.2.1.69 ChunkEncoderSelector

[quickSpinEnumerationNode](#) ChunkEncoderSelector

13.2.1.70 ChunkEncoderStatus

[quickSpinEnumerationNode](#) ChunkEncoderStatus

13.2.1.71 ChunkEncoderValue

[quickSpinIntegerNode](#) ChunkEncoderValue

13.2.1.72 ChunkExposureEndLineStatusAll

[quickSpinIntegerNode](#) ChunkExposureEndLineStatusAll

13.2.1.73 ChunkExposureTime

[quickSpinFloatNode](#) ChunkExposureTime

13.2.1.74 ChunkExposureTimeSelector

[quickSpinEnumerationNode](#) ChunkExposureTimeSelector

13.2.1.75 ChunkFrameID

[quickSpinIntegerNode](#) ChunkFrameID

13.2.1.76 ChunkGain

[quickSpinFloatNode](#) ChunkGain

13.2.1.77 ChunkGainSelector

[quickSpinEnumerationNode](#) ChunkGainSelector

13.2.1.78 ChunkHeight

[quickSpinIntegerNode](#) ChunkHeight

13.2.1.79 ChunkImage

[quickSpinIntegerNode](#) ChunkImage

13.2.1.80 ChunkImageComponent

[quickSpinEnumerationNode](#) ChunkImageComponent

13.2.1.81 ChunkInferenceBoundingBoxResult

[quickSpinRegisterNode](#) ChunkInferenceBoundingBoxResult

13.2.1.82 ChunkInferenceConfidence

[quickSpinFloatNode](#) ChunkInferenceConfidence

13.2.1.83 ChunkInferenceFrameId

[quickSpinIntegerNode](#) ChunkInferenceFrameId

13.2.1.84 ChunkInferenceResult

[quickSpinIntegerNode](#) ChunkInferenceResult

13.2.1.85 ChunkLinePitch

`quickSpinIntegerNode` ChunkLinePitch

13.2.1.86 ChunkLineStatusAll

`quickSpinIntegerNode` ChunkLineStatusAll

13.2.1.87 ChunkModeActive

`quickSpinBooleanNode` ChunkModeActive

13.2.1.88 ChunkOffsetX

`quickSpinIntegerNode` ChunkOffsetX

13.2.1.89 ChunkOffsetY

`quickSpinIntegerNode` ChunkOffsetY

13.2.1.90 ChunkPartSelector

`quickSpinIntegerNode` ChunkPartSelector

13.2.1.91 ChunkPixelDynamicRangeMax

`quickSpinIntegerNode` ChunkPixelDynamicRangeMax

13.2.1.92 ChunkPixelDynamicRangeMin

`quickSpinIntegerNode` ChunkPixelDynamicRangeMin

13.2.1.93 ChunkPixelFormat

[quickSpinEnumerationNode](#) ChunkPixelFormat

13.2.1.94 ChunkRegionID

[quickSpinEnumerationNode](#) ChunkRegionID

13.2.1.95 ChunkScan3dAxisMax

[quickSpinFloatNode](#) ChunkScan3dAxisMax

13.2.1.96 ChunkScan3dAxisMin

[quickSpinFloatNode](#) ChunkScan3dAxisMin

13.2.1.97 ChunkScan3dCoordinateOffset

[quickSpinFloatNode](#) ChunkScan3dCoordinateOffset

13.2.1.98 ChunkScan3dCoordinateReferenceSelector

[quickSpinEnumerationNode](#) ChunkScan3dCoordinateReferenceSelector

13.2.1.99 ChunkScan3dCoordinateReferenceValue

[quickSpinFloatNode](#) ChunkScan3dCoordinateReferenceValue

13.2.1.100 ChunkScan3dCoordinateScale

[quickSpinFloatNode](#) ChunkScan3dCoordinateScale

13.2.1.101 ChunkScan3dCoordinateSelector

`quickSpinEnumerationNode` ChunkScan3dCoordinateSelector

13.2.1.102 ChunkScan3dCoordinateSystem

`quickSpinEnumerationNode` ChunkScan3dCoordinateSystem

13.2.1.103 ChunkScan3dCoordinateSystemReference

`quickSpinEnumerationNode` ChunkScan3dCoordinateSystemReference

13.2.1.104 ChunkScan3dCoordinateTransformSelector

`quickSpinEnumerationNode` ChunkScan3dCoordinateTransformSelector

13.2.1.105 ChunkScan3dDistanceUnit

`quickSpinEnumerationNode` ChunkScan3dDistanceUnit

13.2.1.106 ChunkScan3dInvalidDataFlag

`quickSpinBooleanNode` ChunkScan3dInvalidDataFlag

13.2.1.107 ChunkScan3dInvalidDataValue

`quickSpinFloatNode` ChunkScan3dInvalidDataValue

13.2.1.108 ChunkScan3dOutputMode

`quickSpinEnumerationNode` ChunkScan3dOutputMode

13.2.1.109 ChunkScan3dTransformValue

`quickSpinFloatNode` ChunkScan3dTransformValue

13.2.1.110 ChunkScanLineSelector

`quickSpinIntegerNode` ChunkScanLineSelector

13.2.1.111 ChunkSelector

`quickSpinEnumerationNode` ChunkSelector

13.2.1.112 ChunkSequencerSetActive

`quickSpinIntegerNode` ChunkSequencerSetActive

13.2.1.113 ChunkSerialData

`quickSpinStringNode` ChunkSerialData

13.2.1.114 ChunkSerialDataLength

`quickSpinIntegerNode` ChunkSerialDataLength

13.2.1.115 ChunkSerialReceiveOverflow

`quickSpinBooleanNode` ChunkSerialReceiveOverflow

13.2.1.116 ChunkSourceID

`quickSpinEnumerationNode` ChunkSourceID

13.2.1.117 ChunkStreamChannelID

`quickSpinIntegerNode` ChunkStreamChannelID

13.2.1.118 ChunkTimerSelector

`quickSpinEnumerationNode` ChunkTimerSelector

13.2.1.119 ChunkTimerValue

`quickSpinFloatNode` ChunkTimerValue

13.2.1.120 ChunkTimestamp

`quickSpinIntegerNode` ChunkTimestamp

13.2.1.121 ChunkTimestampLatchValue

`quickSpinIntegerNode` ChunkTimestampLatchValue

13.2.1.122 ChunkTransferBlockID

`quickSpinIntegerNode` ChunkTransferBlockID

13.2.1.123 ChunkTransferQueueCurrentBlockCount

`quickSpinIntegerNode` ChunkTransferQueueCurrentBlockCount

13.2.1.124 ChunkTransferStreamID

`quickSpinEnumerationNode` ChunkTransferStreamID

13.2.1.125 ChunkWidth

`quickSpinIntegerNode` ChunkWidth

13.2.1.126 ClConfiguration

`quickSpinEnumerationNode` ClConfiguration

13.2.1.127 ClTimeSlotsCount

`quickSpinEnumerationNode` ClTimeSlotsCount

13.2.1.128 ColorTransformationEnable

`quickSpinBooleanNode` ColorTransformationEnable

13.2.1.129 ColorTransformationSelector

`quickSpinEnumerationNode` ColorTransformationSelector

13.2.1.130 ColorTransformationValue

`quickSpinFloatNode` ColorTransformationValue

13.2.1.131 ColorTransformationValueSelector

`quickSpinEnumerationNode` ColorTransformationValueSelector

13.2.1.132 CompressionRatio

`quickSpinFloatNode` CompressionRatio

13.2.1.133 CompressionSaturationPriority

`quickSpinEnumerationNode` CompressionSaturationPriority

13.2.1.134 CounterDelay

`quickSpinIntegerNode` CounterDelay

13.2.1.135 CounterDuration

`quickSpinIntegerNode` CounterDuration

13.2.1.136 CounterEventActivation

`quickSpinEnumerationNode` CounterEventActivation

13.2.1.137 CounterEventSource

`quickSpinEnumerationNode` CounterEventSource

13.2.1.138 CounterReset

`quickSpinCommandNode` CounterReset

13.2.1.139 CounterResetActivation

`quickSpinEnumerationNode` CounterResetActivation

13.2.1.140 CounterResetSource

`quickSpinEnumerationNode` CounterResetSource

13.2.1.141 CounterSelector

[quickSpinEnumerationNode](#) CounterSelector

13.2.1.142 CounterStatus

[quickSpinEnumerationNode](#) CounterStatus

13.2.1.143 CounterTriggerActivation

[quickSpinEnumerationNode](#) CounterTriggerActivation

13.2.1.144 CounterTriggerSource

[quickSpinEnumerationNode](#) CounterTriggerSource

13.2.1.145 CounterValue

[quickSpinIntegerNode](#) CounterValue

13.2.1.146 CounterValueAtReset

[quickSpinIntegerNode](#) CounterValueAtReset

13.2.1.147 CxpConnectionSelector

[quickSpinIntegerNode](#) CxpConnectionSelector

13.2.1.148 CxpConnectionTestErrorCount

[quickSpinIntegerNode](#) CxpConnectionTestErrorCount

13.2.1.149 CxpConnectionTestMode

[quickSpinEnumerationNode](#) CxpConnectionTestMode

13.2.1.150 CxpConnectionTestPacketCount

[quickSpinIntegerNode](#) CxpConnectionTestPacketCount

13.2.1.151 CxpLinkConfiguration

[quickSpinEnumerationNode](#) CxpLinkConfiguration

13.2.1.152 CxpLinkConfigurationPreferred

[quickSpinEnumerationNode](#) CxpLinkConfigurationPreferred

13.2.1.153 CxpLinkConfigurationStatus

[quickSpinEnumerationNode](#) CxpLinkConfigurationStatus

13.2.1.154 CxpPoCxpAuto

[quickSpinCommandNode](#) CxpPoCxpAuto

13.2.1.155 CxpPoCxpStatus

[quickSpinEnumerationNode](#) CxpPoCxpStatus

13.2.1.156 CxpPoCxpTripReset

[quickSpinCommandNode](#) CxpPoCxpTripReset

13.2.1.157 CxpPoCxpTurnOff

[quickSpinCommandNode](#) CxpPoCxpTurnOff

13.2.1.158 DecimationHorizontal

[quickSpinIntegerNode](#) DecimationHorizontal

13.2.1.159 DecimationHorizontalMode

[quickSpinEnumerationNode](#) DecimationHorizontalMode

13.2.1.160 DecimationSelector

[quickSpinEnumerationNode](#) DecimationSelector

13.2.1.161 DecimationVertical

[quickSpinIntegerNode](#) DecimationVertical

13.2.1.162 DecimationVerticalMode

[quickSpinEnumerationNode](#) DecimationVerticalMode

13.2.1.163 DefectCorrectionMode

[quickSpinEnumerationNode](#) DefectCorrectionMode

13.2.1.164 DefectCorrectStaticEnable

[quickSpinBooleanNode](#) DefectCorrectStaticEnable

13.2.1.165 DefectTableApply

`quickSpinCommandNode` DefectTableApply

13.2.1.166 DefectTableCoordinateX

`quickSpinIntegerNode` DefectTableCoordinateX

13.2.1.167 DefectTableCoordinateY

`quickSpinIntegerNode` DefectTableCoordinateY

13.2.1.168 DefectTableFactoryRestore

`quickSpinCommandNode` DefectTableFactoryRestore

13.2.1.169 DefectTableIndex

`quickSpinIntegerNode` DefectTableIndex

13.2.1.170 DefectTablePixelCount

`quickSpinIntegerNode` DefectTablePixelCount

13.2.1.171 DefectTableSave

`quickSpinCommandNode` DefectTableSave

13.2.1.172 Deinterlacing

`quickSpinEnumerationNode` Deinterlacing

13.2.1.173 DeviceCharacterSet

[quickSpinEnumerationNode](#) DeviceCharacterSet

13.2.1.174 DeviceClockFrequency

[quickSpinFloatNode](#) DeviceClockFrequency

13.2.1.175 DeviceClockSelector

[quickSpinEnumerationNode](#) DeviceClockSelector

13.2.1.176 DeviceConnectionSelector

[quickSpinIntegerNode](#) DeviceConnectionSelector

13.2.1.177 DeviceConnectionSpeed

[quickSpinIntegerNode](#) DeviceConnectionSpeed

13.2.1.178 DeviceConnectionStatus

[quickSpinEnumerationNode](#) DeviceConnectionStatus

13.2.1.179 DeviceEventChannelCount

[quickSpinIntegerNode](#) DeviceEventChannelCount

13.2.1.180 DeviceFamilyName

[quickSpinStringNode](#) DeviceFamilyName

13.2.1.181 DeviceFeaturePersistenceEnd

[quickSpinCommandNode](#) DeviceFeaturePersistenceEnd

13.2.1.182 DeviceFeaturePersistenceStart

[quickSpinCommandNode](#) DeviceFeaturePersistenceStart

13.2.1.183 DeviceFirmwareVersion

[quickSpinStringNode](#) DeviceFirmwareVersion

13.2.1.184 DeviceGenCPVersionMajor

[quickSpinIntegerNode](#) DeviceGenCPVersionMajor

13.2.1.185 DeviceGenCPVersionMinor

[quickSpinIntegerNode](#) DeviceGenCPVersionMinor

13.2.1.186 DeviceID

[quickSpinStringNode](#) DeviceID

13.2.1.187 DeviceIndicatorMode

[quickSpinEnumerationNode](#) DeviceIndicatorMode

13.2.1.188 DeviceLinkBandwidthReserve

[quickSpinFloatNode](#) DeviceLinkBandwidthReserve

13.2.1.189 DeviceLinkCommandTimeout

`quickSpinFloatNode` DeviceLinkCommandTimeout

13.2.1.190 DeviceLinkConnectionCount

`quickSpinIntegerNode` DeviceLinkConnectionCount

13.2.1.191 DeviceLinkCurrentThroughput

`quickSpinIntegerNode` DeviceLinkCurrentThroughput

13.2.1.192 DeviceLinkHeartbeatMode

`quickSpinEnumerationNode` DeviceLinkHeartbeatMode

13.2.1.193 DeviceLinkHeartbeatTimeout

`quickSpinFloatNode` DeviceLinkHeartbeatTimeout

13.2.1.194 DeviceLinkSelector

`quickSpinIntegerNode` DeviceLinkSelector

13.2.1.195 DeviceLinkSpeed

`quickSpinIntegerNode` DeviceLinkSpeed

13.2.1.196 DeviceLinkThroughputLimit

`quickSpinIntegerNode` DeviceLinkThroughputLimit

13.2.1.197 DeviceLinkThroughputLimitMode

[quickSpinEnumerationNode](#) DeviceLinkThroughputLimitMode

13.2.1.198 DeviceManifestEntrySelector

[quickSpinIntegerNode](#) DeviceManifestEntrySelector

13.2.1.199 DeviceManifestPrimaryURL

[quickSpinStringNode](#) DeviceManifestPrimaryURL

13.2.1.200 DeviceManifestSchemaMajorVersion

[quickSpinIntegerNode](#) DeviceManifestSchemaMajorVersion

13.2.1.201 DeviceManifestSchemaMinorVersion

[quickSpinIntegerNode](#) DeviceManifestSchemaMinorVersion

13.2.1.202 DeviceManifestSecondaryURL

[quickSpinStringNode](#) DeviceManifestSecondaryURL

13.2.1.203 DeviceManifestXMLMajorVersion

[quickSpinIntegerNode](#) DeviceManifestXMLMajorVersion

13.2.1.204 DeviceManifestXMLMinorVersion

[quickSpinIntegerNode](#) DeviceManifestXMLMinorVersion

13.2.1.205 DeviceManifestXMLSubMinorVersion

`quickSpinIntegerNode` DeviceManifestXMLSubMinorVersion

13.2.1.206 DeviceManufacturerInfo

`quickSpinStringNode` DeviceManufacturerInfo

13.2.1.207 DeviceMaxThroughput

`quickSpinIntegerNode` DeviceMaxThroughput

13.2.1.208 DeviceModelName

`quickSpinStringNode` DeviceModelName

13.2.1.209 DevicePowerSupplySelector

`quickSpinEnumerationNode` DevicePowerSupplySelector

13.2.1.210 DeviceRegistersCheck

`quickSpinCommandNode` DeviceRegistersCheck

13.2.1.211 DeviceRegistersEndianness

`quickSpinEnumerationNode` DeviceRegistersEndianness

13.2.1.212 DeviceRegistersStreamingEnd

`quickSpinCommandNode` DeviceRegistersStreamingEnd

13.2.1.213 DeviceRegistersStreamingStart

[quickSpinCommandNode](#) DeviceRegistersStreamingStart

13.2.1.214 DeviceRegistersValid

[quickSpinBooleanNode](#) DeviceRegistersValid

13.2.1.215 DeviceReset

[quickSpinCommandNode](#) DeviceReset

13.2.1.216 DeviceScanType

[quickSpinEnumerationNode](#) DeviceScanType

13.2.1.217 DeviceSerialNumber

[quickSpinStringNode](#) DeviceSerialNumber

13.2.1.218 DeviceSerialPortBaudRate

[quickSpinEnumerationNode](#) DeviceSerialPortBaudRate

13.2.1.219 DeviceSerialPortSelector

[quickSpinEnumerationNode](#) DeviceSerialPortSelector

13.2.1.220 DeviceSFNCVersionMajor

[quickSpinIntegerNode](#) DeviceSFNCVersionMajor

13.2.1.221 DeviceSFNCVersionMinor

[quickSpinIntegerNode](#) DeviceSFNCVersionMinor

13.2.1.222 DeviceSFNCVersionSubMinor

[quickSpinIntegerNode](#) DeviceSFNCVersionSubMinor

13.2.1.223 DeviceStreamChannelCount

[quickSpinIntegerNode](#) DeviceStreamChannelCount

13.2.1.224 DeviceStreamChannelEndianness

[quickSpinEnumerationNode](#) DeviceStreamChannelEndianness

13.2.1.225 DeviceStreamChannelLink

[quickSpinIntegerNode](#) DeviceStreamChannelLink

13.2.1.226 DeviceStreamChannelPacketSize

[quickSpinIntegerNode](#) DeviceStreamChannelPacketSize

13.2.1.227 DeviceStreamChannelSelector

[quickSpinIntegerNode](#) DeviceStreamChannelSelector

13.2.1.228 DeviceStreamChannelType

[quickSpinEnumerationNode](#) DeviceStreamChannelType

13.2.1.229 DeviceTapGeometry

[quickSpinEnumerationNode](#) DeviceTapGeometry

13.2.1.230 DeviceTemperature

[quickSpinFloatNode](#) DeviceTemperature

13.2.1.231 DeviceTemperatureSelector

[quickSpinEnumerationNode](#) DeviceTemperatureSelector

13.2.1.232 DeviceTLType

[quickSpinEnumerationNode](#) DeviceTLType

13.2.1.233 DeviceTLVersionMajor

[quickSpinIntegerNode](#) DeviceTLVersionMajor

13.2.1.234 DeviceTLVersionMinor

[quickSpinIntegerNode](#) DeviceTLVersionMinor

13.2.1.235 DeviceTLVersionSubMinor

[quickSpinIntegerNode](#) DeviceTLVersionSubMinor

13.2.1.236 DeviceType

[quickSpinEnumerationNode](#) DeviceType

13.2.1.237 DeviceUptime

[quickSpinIntegerNode](#) DeviceUptime

13.2.1.238 DeviceUserID

[quickSpinStringNode](#) DeviceUserID

13.2.1.239 DeviceVendorName

[quickSpinStringNode](#) DeviceVendorName

13.2.1.240 DeviceVersion

[quickSpinStringNode](#) DeviceVersion

13.2.1.241 EncoderDivider

[quickSpinIntegerNode](#) EncoderDivider

13.2.1.242 EncoderMode

[quickSpinEnumerationNode](#) EncoderMode

13.2.1.243 EncoderOutputMode

[quickSpinEnumerationNode](#) EncoderOutputMode

13.2.1.244 EncoderReset

[quickSpinCommandNode](#) EncoderReset

13.2.1.245 EncoderResetActivation

[quickSpinEnumerationNode](#) EncoderResetActivation

13.2.1.246 EncoderResetSource

[quickSpinEnumerationNode](#) EncoderResetSource

13.2.1.247 EncoderSelector

[quickSpinEnumerationNode](#) EncoderSelector

13.2.1.248 EncoderSourceA

[quickSpinEnumerationNode](#) EncoderSourceA

13.2.1.249 EncoderSourceB

[quickSpinEnumerationNode](#) EncoderSourceB

13.2.1.250 EncoderStatus

[quickSpinEnumerationNode](#) EncoderStatus

13.2.1.251 EncoderTimeout

[quickSpinFloatNode](#) EncoderTimeout

13.2.1.252 EncoderValue

[quickSpinIntegerNode](#) EncoderValue

13.2.1.253 EncoderValueAtReset

`quickSpinIntegerNode` EncoderValueAtReset

13.2.1.254 EnumerationCount

`quickSpinIntegerNode` EnumerationCount

13.2.1.255 EventAcquisitionEnd

`quickSpinIntegerNode` EventAcquisitionEnd

13.2.1.256 EventAcquisitionEndFrameID

`quickSpinIntegerNode` EventAcquisitionEndFrameID

13.2.1.257 EventAcquisitionEndTimestamp

`quickSpinIntegerNode` EventAcquisitionEndTimestamp

13.2.1.258 EventAcquisitionError

`quickSpinIntegerNode` EventAcquisitionError

13.2.1.259 EventAcquisitionErrorFrameID

`quickSpinIntegerNode` EventAcquisitionErrorFrameID

13.2.1.260 EventAcquisitionErrorTimestamp

`quickSpinIntegerNode` EventAcquisitionErrorTimestamp

13.2.1.261 EventAcquisitionStart

`quickSpinIntegerNode` EventAcquisitionStart

13.2.1.262 EventAcquisitionStartFrameID

`quickSpinIntegerNode` EventAcquisitionStartFrameID

13.2.1.263 EventAcquisitionStartTimestamp

`quickSpinIntegerNode` EventAcquisitionStartTimestamp

13.2.1.264 EventAcquisitionTransferEnd

`quickSpinIntegerNode` EventAcquisitionTransferEnd

13.2.1.265 EventAcquisitionTransferEndFrameID

`quickSpinIntegerNode` EventAcquisitionTransferEndFrameID

13.2.1.266 EventAcquisitionTransferEndTimestamp

`quickSpinIntegerNode` EventAcquisitionTransferEndTimestamp

13.2.1.267 EventAcquisitionTransferStart

`quickSpinIntegerNode` EventAcquisitionTransferStart

13.2.1.268 EventAcquisitionTransferStartFrameID

`quickSpinIntegerNode` EventAcquisitionTransferStartFrameID

13.2.1.269 EventAcquisitionTransferStartTimestamp

`quickSpinIntegerNode` EventAcquisitionTransferStartTimestamp

13.2.1.270 EventAcquisitionTrigger

`quickSpinIntegerNode` EventAcquisitionTrigger

13.2.1.271 EventAcquisitionTriggerFrameID

`quickSpinIntegerNode` EventAcquisitionTriggerFrameID

13.2.1.272 EventAcquisitionTriggerTimestamp

`quickSpinIntegerNode` EventAcquisitionTriggerTimestamp

13.2.1.273 EventActionLate

`quickSpinIntegerNode` EventActionLate

13.2.1.274 EventActionLateFrameID

`quickSpinIntegerNode` EventActionLateFrameID

13.2.1.275 EventActionLateTimestamp

`quickSpinIntegerNode` EventActionLateTimestamp

13.2.1.276 EventCounter0End

`quickSpinIntegerNode` EventCounter0End

13.2.1.277 EventCounter0EndFrameID

`quickSpinIntegerNode` EventCounter0EndFrameID

13.2.1.278 EventCounter0EndTimestamp

`quickSpinIntegerNode` EventCounter0EndTimestamp

13.2.1.279 EventCounter0Start

`quickSpinIntegerNode` EventCounter0Start

13.2.1.280 EventCounter0StartFrameID

`quickSpinIntegerNode` EventCounter0StartFrameID

13.2.1.281 EventCounter0StartTimestamp

`quickSpinIntegerNode` EventCounter0StartTimestamp

13.2.1.282 EventCounter1End

`quickSpinIntegerNode` EventCounter1End

13.2.1.283 EventCounter1EndFrameID

`quickSpinIntegerNode` EventCounter1EndFrameID

13.2.1.284 EventCounter1EndTimestamp

`quickSpinIntegerNode` EventCounter1EndTimestamp

13.2.1.285 EventCounter1Start

[quickSpinIntegerNode](#) EventCounter1Start

13.2.1.286 EventCounter1StartFrameID

[quickSpinIntegerNode](#) EventCounter1StartFrameID

13.2.1.287 EventCounter1StartTimestamp

[quickSpinIntegerNode](#) EventCounter1StartTimestamp

13.2.1.288 EventEncoder0Restarted

[quickSpinIntegerNode](#) EventEncoder0Restarted

13.2.1.289 EventEncoder0RestartedFrameID

[quickSpinIntegerNode](#) EventEncoder0RestartedFrameID

13.2.1.290 EventEncoder0RestartedTimestamp

[quickSpinIntegerNode](#) EventEncoder0RestartedTimestamp

13.2.1.291 EventEncoder0Stopped

[quickSpinIntegerNode](#) EventEncoder0Stopped

13.2.1.292 EventEncoder0StoppedFrameID

[quickSpinIntegerNode](#) EventEncoder0StoppedFrameID

13.2.1.293 EventEncoder0StoppedTimestamp

`quickSpinIntegerNode` EventEncoder0StoppedTimestamp

13.2.1.294 EventEncoder1Restarted

`quickSpinIntegerNode` EventEncoder1Restarted

13.2.1.295 EventEncoder1RestartedFrameID

`quickSpinIntegerNode` EventEncoder1RestartedFrameID

13.2.1.296 EventEncoder1RestartedTimestamp

`quickSpinIntegerNode` EventEncoder1RestartedTimestamp

13.2.1.297 EventEncoder1Stopped

`quickSpinIntegerNode` EventEncoder1Stopped

13.2.1.298 EventEncoder1StoppedFrameID

`quickSpinIntegerNode` EventEncoder1StoppedFrameID

13.2.1.299 EventEncoder1StoppedTimestamp

`quickSpinIntegerNode` EventEncoder1StoppedTimestamp

13.2.1.300 EventError

`quickSpinIntegerNode` EventError

13.2.1.301 EventErrorCode

`quickSpinIntegerNode` EventErrorCode

13.2.1.302 EventErrorFrameID

`quickSpinIntegerNode` EventErrorFrameID

13.2.1.303 EventErrorTimestamp

`quickSpinIntegerNode` EventErrorTimestamp

13.2.1.304 EventExposureEnd

`quickSpinIntegerNode` EventExposureEnd

13.2.1.305 EventExposureEndFrameID

`quickSpinIntegerNode` EventExposureEndFrameID

13.2.1.306 EventExposureEndTimestamp

`quickSpinIntegerNode` EventExposureEndTimestamp

13.2.1.307 EventExposureStart

`quickSpinIntegerNode` EventExposureStart

13.2.1.308 EventExposureStartFrameID

`quickSpinIntegerNode` EventExposureStartFrameID

13.2.1.309 EventExposureStartTimestamp

[quickSpinIntegerNode](#) EventExposureStartTimestamp

13.2.1.310 EventFrameBurstEnd

[quickSpinIntegerNode](#) EventFrameBurstEnd

13.2.1.311 EventFrameBurstEndFrameID

[quickSpinIntegerNode](#) EventFrameBurstEndFrameID

13.2.1.312 EventFrameBurstEndTimestamp

[quickSpinIntegerNode](#) EventFrameBurstEndTimestamp

13.2.1.313 EventFrameBurstStart

[quickSpinIntegerNode](#) EventFrameBurstStart

13.2.1.314 EventFrameBurstStartFrameID

[quickSpinIntegerNode](#) EventFrameBurstStartFrameID

13.2.1.315 EventFrameBurstStartTimestamp

[quickSpinIntegerNode](#) EventFrameBurstStartTimestamp

13.2.1.316 EventFrameEnd

[quickSpinIntegerNode](#) EventFrameEnd

13.2.1.317 EventFrameEndFrameID

[quickSpinIntegerNode](#) EventFrameEndFrameID

13.2.1.318 EventFrameEndTimestamp

[quickSpinIntegerNode](#) EventFrameEndTimestamp

13.2.1.319 EventFrameStart

[quickSpinIntegerNode](#) EventFrameStart

13.2.1.320 EventFrameStartFrameID

[quickSpinIntegerNode](#) EventFrameStartFrameID

13.2.1.321 EventFrameStartTimestamp

[quickSpinIntegerNode](#) EventFrameStartTimestamp

13.2.1.322 EventFrameTransferEnd

[quickSpinIntegerNode](#) EventFrameTransferEnd

13.2.1.323 EventFrameTransferEndFrameID

[quickSpinIntegerNode](#) EventFrameTransferEndFrameID

13.2.1.324 EventFrameTransferEndTimestamp

[quickSpinIntegerNode](#) EventFrameTransferEndTimestamp

13.2.1.325 EventFrameTransferStart

`quickSpinIntegerNode` EventFrameTransferStart

13.2.1.326 EventFrameTransferStartFrameID

`quickSpinIntegerNode` EventFrameTransferStartFrameID

13.2.1.327 EventFrameTransferStartTimestamp

`quickSpinIntegerNode` EventFrameTransferStartTimestamp

13.2.1.328 EventFrameTrigger

`quickSpinIntegerNode` EventFrameTrigger

13.2.1.329 EventFrameTriggerFrameID

`quickSpinIntegerNode` EventFrameTriggerFrameID

13.2.1.330 EventFrameTriggerTimestamp

`quickSpinIntegerNode` EventFrameTriggerTimestamp

13.2.1.331 EventLine0AnyEdge

`quickSpinIntegerNode` EventLine0AnyEdge

13.2.1.332 EventLine0AnyEdgeFrameID

`quickSpinIntegerNode` EventLine0AnyEdgeFrameID

13.2.1.333 EventLine0AnyEdgeTimestamp

`quickSpinIntegerNode` EventLine0AnyEdgeTimestamp

13.2.1.334 EventLine0FallingEdge

`quickSpinIntegerNode` EventLine0FallingEdge

13.2.1.335 EventLine0FallingEdgeFrameID

`quickSpinIntegerNode` EventLine0FallingEdgeFrameID

13.2.1.336 EventLine0FallingEdgeTimestamp

`quickSpinIntegerNode` EventLine0FallingEdgeTimestamp

13.2.1.337 EventLine0RisingEdge

`quickSpinIntegerNode` EventLine0RisingEdge

13.2.1.338 EventLine0RisingEdgeFrameID

`quickSpinIntegerNode` EventLine0RisingEdgeFrameID

13.2.1.339 EventLine0RisingEdgeTimestamp

`quickSpinIntegerNode` EventLine0RisingEdgeTimestamp

13.2.1.340 EventLine1AnyEdge

`quickSpinIntegerNode` EventLine1AnyEdge

13.2.1.341 EventLine1AnyEdgeFrameID

`quickSpinIntegerNode` EventLine1AnyEdgeFrameID

13.2.1.342 EventLine1AnyEdgeTimestamp

`quickSpinIntegerNode` EventLine1AnyEdgeTimestamp

13.2.1.343 EventLine1FallingEdge

`quickSpinIntegerNode` EventLine1FallingEdge

13.2.1.344 EventLine1FallingEdgeFrameID

`quickSpinIntegerNode` EventLine1FallingEdgeFrameID

13.2.1.345 EventLine1FallingEdgeTimestamp

`quickSpinIntegerNode` EventLine1FallingEdgeTimestamp

13.2.1.346 EventLine1RisingEdge

`quickSpinIntegerNode` EventLine1RisingEdge

13.2.1.347 EventLine1RisingEdgeFrameID

`quickSpinIntegerNode` EventLine1RisingEdgeFrameID

13.2.1.348 EventLine1RisingEdgeTimestamp

`quickSpinIntegerNode` EventLine1RisingEdgeTimestamp

13.2.1.349 EventLinkSpeedChange

[quickSpinIntegerNode](#) EventLinkSpeedChange

13.2.1.350 EventLinkSpeedChangeFrameID

[quickSpinIntegerNode](#) EventLinkSpeedChangeFrameID

13.2.1.351 EventLinkSpeedChangeTimestamp

[quickSpinIntegerNode](#) EventLinkSpeedChangeTimestamp

13.2.1.352 EventLinkTrigger0

[quickSpinIntegerNode](#) EventLinkTrigger0

13.2.1.353 EventLinkTrigger0FrameID

[quickSpinIntegerNode](#) EventLinkTrigger0FrameID

13.2.1.354 EventLinkTrigger0Timestamp

[quickSpinIntegerNode](#) EventLinkTrigger0Timestamp

13.2.1.355 EventLinkTrigger1

[quickSpinIntegerNode](#) EventLinkTrigger1

13.2.1.356 EventLinkTrigger1FrameID

[quickSpinIntegerNode](#) EventLinkTrigger1FrameID

13.2.1.357 EventLinkTrigger1Timestamp

`quickSpinIntegerNode` EventLinkTrigger1Timestamp

13.2.1.358 EventNotification

`quickSpinEnumerationNode` EventNotification

13.2.1.359 EventSelector

`quickSpinEnumerationNode` EventSelector

13.2.1.360 EventSequencerSetChange

`quickSpinIntegerNode` EventSequencerSetChange

13.2.1.361 EventSequencerSetChangeFrameID

`quickSpinIntegerNode` EventSequencerSetChangeFrameID

13.2.1.362 EventSequencerSetChangeTimestamp

`quickSpinIntegerNode` EventSequencerSetChangeTimestamp

13.2.1.363 EventSerialData

`quickSpinStringNode` EventSerialData

13.2.1.364 EventSerialDataLength

`quickSpinIntegerNode` EventSerialDataLength

13.2.1.365 EventSerialPortReceive

`quickSpinIntegerNode` EventSerialPortReceive

13.2.1.366 EventSerialPortReceiveTimestamp

`quickSpinIntegerNode` EventSerialPortReceiveTimestamp

13.2.1.367 EventSerialReceiveOverflow

`quickSpinBooleanNode` EventSerialReceiveOverflow

13.2.1.368 EventStream0TransferBlockEnd

`quickSpinIntegerNode` EventStream0TransferBlockEnd

13.2.1.369 EventStream0TransferBlockEndFrameID

`quickSpinIntegerNode` EventStream0TransferBlockEndFrameID

13.2.1.370 EventStream0TransferBlockEndTimestamp

`quickSpinIntegerNode` EventStream0TransferBlockEndTimestamp

13.2.1.371 EventStream0TransferBlockStart

`quickSpinIntegerNode` EventStream0TransferBlockStart

13.2.1.372 EventStream0TransferBlockStartFrameID

`quickSpinIntegerNode` EventStream0TransferBlockStartFrameID

13.2.1.373 EventStream0TransferBlockStartTimestamp

`quickSpinIntegerNode` EventStream0TransferBlockStartTimestamp

13.2.1.374 EventStream0TransferBlockTrigger

`quickSpinIntegerNode` EventStream0TransferBlockTrigger

13.2.1.375 EventStream0TransferBlockTriggerFrameID

`quickSpinIntegerNode` EventStream0TransferBlockTriggerFrameID

13.2.1.376 EventStream0TransferBlockTriggerTimestamp

`quickSpinIntegerNode` EventStream0TransferBlockTriggerTimestamp

13.2.1.377 EventStream0TransferBurstEnd

`quickSpinIntegerNode` EventStream0TransferBurstEnd

13.2.1.378 EventStream0TransferBurstEndFrameID

`quickSpinIntegerNode` EventStream0TransferBurstEndFrameID

13.2.1.379 EventStream0TransferBurstEndTimestamp

`quickSpinIntegerNode` EventStream0TransferBurstEndTimestamp

13.2.1.380 EventStream0TransferBurstStart

`quickSpinIntegerNode` EventStream0TransferBurstStart

13.2.1.381 EventStream0TransferBurstStartFrameID

`quickSpinIntegerNode` EventStream0TransferBurstStartFrameID

13.2.1.382 EventStream0TransferBurstStartTimestamp

`quickSpinIntegerNode` EventStream0TransferBurstStartTimestamp

13.2.1.383 EventStream0TransferEnd

`quickSpinIntegerNode` EventStream0TransferEnd

13.2.1.384 EventStream0TransferEndFrameID

`quickSpinIntegerNode` EventStream0TransferEndFrameID

13.2.1.385 EventStream0TransferEndTimestamp

`quickSpinIntegerNode` EventStream0TransferEndTimestamp

13.2.1.386 EventStream0TransferOverflow

`quickSpinIntegerNode` EventStream0TransferOverflow

13.2.1.387 EventStream0TransferOverflowFrameID

`quickSpinIntegerNode` EventStream0TransferOverflowFrameID

13.2.1.388 EventStream0TransferOverflowTimestamp

`quickSpinIntegerNode` EventStream0TransferOverflowTimestamp

13.2.1.389 EventStream0TransferPause

[quickSpinIntegerNode](#) EventStream0TransferPause

13.2.1.390 EventStream0TransferPauseFrameID

[quickSpinIntegerNode](#) EventStream0TransferPauseFrameID

13.2.1.391 EventStream0TransferPauseTimestamp

[quickSpinIntegerNode](#) EventStream0TransferPauseTimestamp

13.2.1.392 EventStream0TransferResume

[quickSpinIntegerNode](#) EventStream0TransferResume

13.2.1.393 EventStream0TransferResumeFrameID

[quickSpinIntegerNode](#) EventStream0TransferResumeFrameID

13.2.1.394 EventStream0TransferResumeTimestamp

[quickSpinIntegerNode](#) EventStream0TransferResumeTimestamp

13.2.1.395 EventStream0TransferStart

[quickSpinIntegerNode](#) EventStream0TransferStart

13.2.1.396 EventStream0TransferStartFrameID

[quickSpinIntegerNode](#) EventStream0TransferStartFrameID

13.2.1.397 EventStream0TransferStartTimestamp

`quickSpinIntegerNode` EventStream0TransferStartTimestamp

13.2.1.398 EventTest

`quickSpinIntegerNode` EventTest

13.2.1.399 EventTestTimestamp

`quickSpinIntegerNode` EventTestTimestamp

13.2.1.400 EventTimer0End

`quickSpinIntegerNode` EventTimer0End

13.2.1.401 EventTimer0EndFrameID

`quickSpinIntegerNode` EventTimer0EndFrameID

13.2.1.402 EventTimer0EndTimestamp

`quickSpinIntegerNode` EventTimer0EndTimestamp

13.2.1.403 EventTimer0Start

`quickSpinIntegerNode` EventTimer0Start

13.2.1.404 EventTimer0StartFrameID

`quickSpinIntegerNode` EventTimer0StartFrameID

13.2.1.405 EventTimer0StartTimestamp

[quickSpinIntegerNode](#) EventTimer0StartTimestamp

13.2.1.406 EventTimer1End

[quickSpinIntegerNode](#) EventTimer1End

13.2.1.407 EventTimer1EndFrameID

[quickSpinIntegerNode](#) EventTimer1EndFrameID

13.2.1.408 EventTimer1EndTimestamp

[quickSpinIntegerNode](#) EventTimer1EndTimestamp

13.2.1.409 EventTimer1Start

[quickSpinIntegerNode](#) EventTimer1Start

13.2.1.410 EventTimer1StartFrameID

[quickSpinIntegerNode](#) EventTimer1StartFrameID

13.2.1.411 EventTimer1StartTimestamp

[quickSpinIntegerNode](#) EventTimer1StartTimestamp

13.2.1.412 ExposureActiveMode

[quickSpinEnumerationNode](#) ExposureActiveMode

13.2.1.413 ExposureAuto

[quickSpinEnumerationNode](#) ExposureAuto

13.2.1.414 ExposureMode

[quickSpinEnumerationNode](#) ExposureMode

13.2.1.415 ExposureTime

[quickSpinFloatNode](#) ExposureTime

13.2.1.416 ExposureTimeMode

[quickSpinEnumerationNode](#) ExposureTimeMode

13.2.1.417 ExposureTimeSelector

[quickSpinEnumerationNode](#) ExposureTimeSelector

13.2.1.418 FactoryReset

[quickSpinCommandNode](#) FactoryReset

13.2.1.419 FileAccessBuffer

[quickSpinRegisterNode](#) FileAccessBuffer

13.2.1.420 FileAccessLength

[quickSpinIntegerNode](#) FileAccessLength

13.2.1.421 FileAccessOffset

`quickSpinIntegerNode` FileAccessOffset

13.2.1.422 FileOpenMode

`quickSpinEnumerationNode` FileOpenMode

13.2.1.423 FileOperationExecute

`quickSpinCommandNode` FileOperationExecute

13.2.1.424 FileOperationResult

`quickSpinIntegerNode` FileOperationResult

13.2.1.425 FileOperationSelector

`quickSpinEnumerationNode` FileOperationSelector

13.2.1.426 FileOperationStatus

`quickSpinEnumerationNode` FileOperationStatus

13.2.1.427 FileSelector

`quickSpinEnumerationNode` FileSelector

13.2.1.428 FileSize

`quickSpinIntegerNode` FileSize

13.2.1.429 Gain

[quickSpinFloatNode](#) Gain

13.2.1.430 GainAuto

[quickSpinEnumerationNode](#) GainAuto

13.2.1.431 GainAutoBalance

[quickSpinEnumerationNode](#) GainAutoBalance

13.2.1.432 GainSelector

[quickSpinEnumerationNode](#) GainSelector

13.2.1.433 Gamma

[quickSpinFloatNode](#) Gamma

13.2.1.434 GammaEnable

[quickSpinBooleanNode](#) GammaEnable

13.2.1.435 GevActiveLinkCount

[quickSpinIntegerNode](#) GevActiveLinkCount

13.2.1.436 GevCCP

[quickSpinEnumerationNode](#) GevCCP

13.2.1.437 GevCurrentDefaultGateway

`quickSpinIntegerNode` GevCurrentDefaultGateway

13.2.1.438 GevCurrentIPAddress

`quickSpinIntegerNode` GevCurrentIPAddress

13.2.1.439 GevCurrentIPConfigurationDHCP

`quickSpinBooleanNode` GevCurrentIPConfigurationDHCP

13.2.1.440 GevCurrentIPConfigurationLLA

`quickSpinBooleanNode` GevCurrentIPConfigurationLLA

13.2.1.441 GevCurrentIPConfigurationPersistentIP

`quickSpinBooleanNode` GevCurrentIPConfigurationPersistentIP

13.2.1.442 GevCurrentPhysicalLinkConfiguration

`quickSpinEnumerationNode` GevCurrentPhysicalLinkConfiguration

13.2.1.443 GevCurrentSubnetMask

`quickSpinIntegerNode` GevCurrentSubnetMask

13.2.1.444 GevDiscoveryAckDelay

`quickSpinIntegerNode` GevDiscoveryAckDelay

13.2.1.445 GevFirstURL

`quickSpinStringNode` `GevFirstURL`

13.2.1.446 GevGVCPExtendedStatusCodes

`quickSpinBooleanNode` `GevGVCPExtendedStatusCodes`

13.2.1.447 GevGVCPExtendedStatusCodesSelector

`quickSpinEnumerationNode` `GevGVCPExtendedStatusCodesSelector`

13.2.1.448 GevGVCPHeartbeatDisable

`quickSpinBooleanNode` `GevGVCPHeartbeatDisable`

13.2.1.449 GevGVCPPendingAck

`quickSpinBooleanNode` `GevGVCPPendingAck`

13.2.1.450 GevGVCPPendingTimeout

`quickSpinIntegerNode` `GevGVCPPendingTimeout`

13.2.1.451 GevGVSPExtendedIDMode

`quickSpinEnumerationNode` `GevGVSPExtendedIDMode`

13.2.1.452 GevHeartbeatTimeout

`quickSpinIntegerNode` `GevHeartbeatTimeout`

13.2.1.453 GevIEEE1588

`quickSpinBooleanNode` `GevIEEE1588`

13.2.1.454 GevIEEE1588ClockAccuracy

`quickSpinEnumerationNode` `GevIEEE1588ClockAccuracy`

13.2.1.455 GevIEEE1588Mode

`quickSpinEnumerationNode` `GevIEEE1588Mode`

13.2.1.456 GevIEEE1588Status

`quickSpinEnumerationNode` `GevIEEE1588Status`

13.2.1.457 GevInterfaceSelector

`quickSpinIntegerNode` `GevInterfaceSelector`

13.2.1.458 GevIPConfigurationStatus

`quickSpinEnumerationNode` `GevIPConfigurationStatus`

13.2.1.459 GevMACAddress

`quickSpinIntegerNode` `GevMACAddress`

13.2.1.460 GevMCDA

`quickSpinIntegerNode` `GevMCDA`

13.2.1.461 GevMCPHostPort

[quickSpinIntegerNode](#) GevMCPHostPort

13.2.1.462 GevMCRC

[quickSpinIntegerNode](#) GevMCRC

13.2.1.463 GevMCSP

[quickSpinIntegerNode](#) GevMCSP

13.2.1.464 GevMCTT

[quickSpinIntegerNode](#) GevMCTT

13.2.1.465 GevNumberOfInterfaces

[quickSpinIntegerNode](#) GevNumberOfInterfaces

13.2.1.466 GevPAUSEFrameReception

[quickSpinBooleanNode](#) GevPAUSEFrameReception

13.2.1.467 GevPAUSEFrameTransmission

[quickSpinBooleanNode](#) GevPAUSEFrameTransmission

13.2.1.468 GevPersistentDefaultGateway

[quickSpinIntegerNode](#) GevPersistentDefaultGateway

13.2.1.469 GevPersistentIPAddress

`quickSpinIntegerNode` GevPersistentIPAddress

13.2.1.470 GevPersistentSubnetMask

`quickSpinIntegerNode` GevPersistentSubnetMask

13.2.1.471 GevPhysicalLinkConfiguration

`quickSpinEnumerationNode` GevPhysicalLinkConfiguration

13.2.1.472 GevPrimaryApplicationIPAddress

`quickSpinIntegerNode` GevPrimaryApplicationIPAddress

13.2.1.473 GevPrimaryApplicationSocket

`quickSpinIntegerNode` GevPrimaryApplicationSocket

13.2.1.474 GevPrimaryApplicationSwitchoverKey

`quickSpinIntegerNode` GevPrimaryApplicationSwitchoverKey

13.2.1.475 GevSCCFGAllInTransmission

`quickSpinBooleanNode` GevSCCFGAllInTransmission

13.2.1.476 GevSCCFGExtendedChunkData

`quickSpinBooleanNode` GevSCCFGExtendedChunkData

13.2.1.477 GevSCCFGPacketResendDestination

[quickSpinBooleanNode](#) GevSCCFGPacketResendDestination

13.2.1.478 GevSCCFGUnconditionalStreaming

[quickSpinBooleanNode](#) GevSCCFGUnconditionalStreaming

13.2.1.479 GevSCDA

[quickSpinIntegerNode](#) GevSCDA

13.2.1.480 GevSCPD

[quickSpinIntegerNode](#) GevSCPD

13.2.1.481 GevSCPDirection

[quickSpinIntegerNode](#) GevSCPDirection

13.2.1.482 GevSCPHostPort

[quickSpinIntegerNode](#) GevSCPHostPort

13.2.1.483 GevSCPInterfaceIndex

[quickSpinIntegerNode](#) GevSCPInterfaceIndex

13.2.1.484 GevSCPSBigEndian

[quickSpinBooleanNode](#) GevSCPSBigEndian

13.2.1.485 GevSCPSDoNotFragment

[quickSpinBooleanNode](#) GevSCPSDoNotFragment

13.2.1.486 GevSCPSFireTestPacket

[quickSpinBooleanNode](#) GevSCPSFireTestPacket

13.2.1.487 GevSCPSPacketSize

[quickSpinIntegerNode](#) GevSCPSPacketSize

13.2.1.488 GevSCSP

[quickSpinIntegerNode](#) GevSCSP

13.2.1.489 GevSCZoneConfigurationLock

[quickSpinBooleanNode](#) GevSCZoneConfigurationLock

13.2.1.490 GevSCZoneCount

[quickSpinIntegerNode](#) GevSCZoneCount

13.2.1.491 GevSCZoneDirectionAll

[quickSpinIntegerNode](#) GevSCZoneDirectionAll

13.2.1.492 GevSecondURL

[quickSpinStringNode](#) GevSecondURL

13.2.1.493 GevStreamChannelSelector

[quickSpinIntegerNode](#) GevStreamChannelSelector

13.2.1.494 GevSupportedOption

[quickSpinBooleanNode](#) GevSupportedOption

13.2.1.495 GevSupportedOptionSelector

[quickSpinEnumerationNode](#) GevSupportedOptionSelector

13.2.1.496 GevTimestampTickFrequency

[quickSpinIntegerNode](#) GevTimestampTickFrequency

13.2.1.497 GuiXmlManifestAddress

[quickSpinIntegerNode](#) GuiXmlManifestAddress

13.2.1.498 Height

[quickSpinIntegerNode](#) Height

13.2.1.499 HeightMax

[quickSpinIntegerNode](#) HeightMax

13.2.1.500 ImageComponentEnable

[quickSpinBooleanNode](#) ImageComponentEnable

13.2.1.501 ImageComponentSelector

`quickSpinEnumerationNode` ImageComponentSelector

13.2.1.502 ImageCompressionBitrate

`quickSpinFloatNode` ImageCompressionBitrate

13.2.1.503 ImageCompressionJPEGFormatOption

`quickSpinEnumerationNode` ImageCompressionJPEGFormatOption

13.2.1.504 ImageCompressionMode

`quickSpinEnumerationNode` ImageCompressionMode

13.2.1.505 ImageCompressionQuality

`quickSpinIntegerNode` ImageCompressionQuality

13.2.1.506 ImageCompressionRateOption

`quickSpinEnumerationNode` ImageCompressionRateOption

13.2.1.507 IspEnable

`quickSpinBooleanNode` IspEnable

13.2.1.508 LineFilterWidth

`quickSpinFloatNode` LineFilterWidth

13.2.1.509 LineFormat

[quickSpinEnumerationNode](#) LineFormat

13.2.1.510 LineInputFilterSelector

[quickSpinEnumerationNode](#) LineInputFilterSelector

13.2.1.511 LineInverter

[quickSpinBooleanNode](#) LineInverter

13.2.1.512 LineMode

[quickSpinEnumerationNode](#) LineMode

13.2.1.513 LinePitch

[quickSpinIntegerNode](#) LinePitch

13.2.1.514 LineSelector

[quickSpinEnumerationNode](#) LineSelector

13.2.1.515 LineSource

[quickSpinEnumerationNode](#) LineSource

13.2.1.516 LineStatus

[quickSpinBooleanNode](#) LineStatus

13.2.1.517 LineStatusAll

`quickSpinIntegerNode` LineStatusAll

13.2.1.518 LinkErrorCount

`quickSpinIntegerNode` LinkErrorCount

13.2.1.519 LinkUptime

`quickSpinIntegerNode` LinkUptime

13.2.1.520 LogicBlockLUTInputActivation

`quickSpinEnumerationNode` LogicBlockLUTInputActivation

13.2.1.521 LogicBlockLUTInputSelector

`quickSpinEnumerationNode` LogicBlockLUTInputSelector

13.2.1.522 LogicBlockLUTInputSource

`quickSpinEnumerationNode` LogicBlockLUTInputSource

13.2.1.523 LogicBlockLUTOutputValue

`quickSpinBooleanNode` LogicBlockLUTOutputValue

13.2.1.524 LogicBlockLUTOutputValueAll

`quickSpinIntegerNode` LogicBlockLUTOutputValueAll

13.2.1.525 LogicBlockLUTRowIndex

[quickSpinIntegerNode](#) LogicBlockLUTRowIndex

13.2.1.526 LogicBlockLUTSelector

[quickSpinEnumerationNode](#) LogicBlockLUTSelector

13.2.1.527 LogicBlockSelector

[quickSpinEnumerationNode](#) LogicBlockSelector

13.2.1.528 LUTEnable

[quickSpinBooleanNode](#) LUTEnable

13.2.1.529 LUTIndex

[quickSpinIntegerNode](#) LUTIndex

13.2.1.530 LUTSelector

[quickSpinEnumerationNode](#) LUTSelector

13.2.1.531 LUTValue

[quickSpinIntegerNode](#) LUTValue

13.2.1.532 LUTValueAll

[quickSpinRegisterNode](#) LUTValueAll

13.2.1.533 MaxDeviceResetTime

[quickSpinIntegerNode](#) MaxDeviceResetTime

13.2.1.534 OffsetX

[quickSpinIntegerNode](#) OffsetX

13.2.1.535 OffsetY

[quickSpinIntegerNode](#) OffsetY

13.2.1.536 PacketResendRequestCount

[quickSpinIntegerNode](#) PacketResendRequestCount

13.2.1.537 PayloadSize

[quickSpinIntegerNode](#) PayloadSize

13.2.1.538 PixelColorFilter

[quickSpinEnumerationNode](#) PixelColorFilter

13.2.1.539 PixelDynamicRangeMax

[quickSpinIntegerNode](#) PixelDynamicRangeMax

13.2.1.540 PixelDynamicRangeMin

[quickSpinIntegerNode](#) PixelDynamicRangeMin

13.2.1.541 PixelFormat

[quickSpinEnumerationNode](#) PixelFormat

13.2.1.542 PixelFormatInfoID

[quickSpinIntegerNode](#) PixelFormatInfoID

13.2.1.543 PixelFormatInfoSelector

[quickSpinEnumerationNode](#) PixelFormatInfoSelector

13.2.1.544 PixelSize

[quickSpinEnumerationNode](#) PixelSize

13.2.1.545 PowerSupplyCurrent

[quickSpinFloatNode](#) PowerSupplyCurrent

13.2.1.546 PowerSupplyVoltage

[quickSpinFloatNode](#) PowerSupplyVoltage

13.2.1.547 RegionDestination

[quickSpinEnumerationNode](#) RegionDestination

13.2.1.548 RegionMode

[quickSpinEnumerationNode](#) RegionMode

13.2.1.549 RegionSelector

`quickSpinEnumerationNode` RegionSelector

13.2.1.550 ReverseX

`quickSpinBooleanNode` ReverseX

13.2.1.551 ReverseY

`quickSpinBooleanNode` ReverseY

13.2.1.552 RgbTransformLightSource

`quickSpinEnumerationNode` RgbTransformLightSource

13.2.1.553 Saturation

`quickSpinFloatNode` Saturation

13.2.1.554 SaturationEnable

`quickSpinBooleanNode` SaturationEnable

13.2.1.555 Scan3dAxisMax

`quickSpinFloatNode` Scan3dAxisMax

13.2.1.556 Scan3dAxisMin

`quickSpinFloatNode` Scan3dAxisMin

13.2.1.557 Scan3dCoordinateOffset

`quickSpinFloatNode` `Scan3dCoordinateOffset`

13.2.1.558 Scan3dCoordinateReferenceSelector

`quickSpinEnumerationNode` `Scan3dCoordinateReferenceSelector`

13.2.1.559 Scan3dCoordinateReferenceValue

`quickSpinFloatNode` `Scan3dCoordinateReferenceValue`

13.2.1.560 Scan3dCoordinateScale

`quickSpinFloatNode` `Scan3dCoordinateScale`

13.2.1.561 Scan3dCoordinateSelector

`quickSpinEnumerationNode` `Scan3dCoordinateSelector`

13.2.1.562 Scan3dCoordinateSystem

`quickSpinEnumerationNode` `Scan3dCoordinateSystem`

13.2.1.563 Scan3dCoordinateSystemReference

`quickSpinEnumerationNode` `Scan3dCoordinateSystemReference`

13.2.1.564 Scan3dCoordinateTransformSelector

`quickSpinEnumerationNode` `Scan3dCoordinateTransformSelector`

13.2.1.565 Scan3dDistanceUnit

`quickSpinEnumerationNode` Scan3dDistanceUnit

13.2.1.566 Scan3dInvalidDataFlag

`quickSpinBooleanNode` Scan3dInvalidDataFlag

13.2.1.567 Scan3dInvalidDataValue

`quickSpinFloatNode` Scan3dInvalidDataValue

13.2.1.568 Scan3dOutputMode

`quickSpinEnumerationNode` Scan3dOutputMode

13.2.1.569 Scan3dTransformValue

`quickSpinFloatNode` Scan3dTransformValue

13.2.1.570 SensorDescription

`quickSpinStringNode` SensorDescription

13.2.1.571 SensorDigitizationTaps

`quickSpinEnumerationNode` SensorDigitizationTaps

13.2.1.572 SensorHeight

`quickSpinIntegerNode` SensorHeight

13.2.1.573 SensorShutterMode

`quickSpinEnumerationNode` SensorShutterMode

13.2.1.574 SensorTaps

`quickSpinEnumerationNode` SensorTaps

13.2.1.575 SensorWidth

`quickSpinIntegerNode` SensorWidth

13.2.1.576 SequencerConfigurationMode

`quickSpinEnumerationNode` SequencerConfigurationMode

13.2.1.577 SequencerConfigurationValid

`quickSpinEnumerationNode` SequencerConfigurationValid

13.2.1.578 SequencerFeatureEnable

`quickSpinBooleanNode` SequencerFeatureEnable

13.2.1.579 SequencerMode

`quickSpinEnumerationNode` SequencerMode

13.2.1.580 SequencerPathSelector

`quickSpinIntegerNode` SequencerPathSelector

13.2.1.581 SequencerSetActive

[quickSpinIntegerNode](#) SequencerSetActive

13.2.1.582 SequencerSetLoad

[quickSpinCommandNode](#) SequencerSetLoad

13.2.1.583 SequencerSetNext

[quickSpinIntegerNode](#) SequencerSetNext

13.2.1.584 SequencerSetSave

[quickSpinCommandNode](#) SequencerSetSave

13.2.1.585 SequencerSetSelector

[quickSpinIntegerNode](#) SequencerSetSelector

13.2.1.586 SequencerSetStart

[quickSpinIntegerNode](#) SequencerSetStart

13.2.1.587 SequencerSetValid

[quickSpinEnumerationNode](#) SequencerSetValid

13.2.1.588 SequencerTriggerActivation

[quickSpinEnumerationNode](#) SequencerTriggerActivation

13.2.1.589 SequencerTriggerSource

[quickSpinEnumerationNode](#) SequencerTriggerSource

13.2.1.590 SerialPortBaudRate

[quickSpinEnumerationNode](#) SerialPortBaudRate

13.2.1.591 SerialPortDataBits

[quickSpinIntegerNode](#) SerialPortDataBits

13.2.1.592 SerialPortParity

[quickSpinEnumerationNode](#) SerialPortParity

13.2.1.593 SerialPortSelector

[quickSpinEnumerationNode](#) SerialPortSelector

13.2.1.594 SerialPortSource

[quickSpinEnumerationNode](#) SerialPortSource

13.2.1.595 SerialPortStopBits

[quickSpinEnumerationNode](#) SerialPortStopBits

13.2.1.596 SerialReceiveFramingErrorCount

[quickSpinIntegerNode](#) SerialReceiveFramingErrorCount

13.2.1.597 SerialReceiveParityErrorCount

`quickSpinIntegerNode` SerialReceiveParityErrorCount

13.2.1.598 SerialReceiveQueueClear

`quickSpinCommandNode` SerialReceiveQueueClear

13.2.1.599 SerialReceiveQueueCurrentCharacterCount

`quickSpinIntegerNode` SerialReceiveQueueCurrentCharacterCount

13.2.1.600 SerialReceiveQueueMaxCharacterCount

`quickSpinIntegerNode` SerialReceiveQueueMaxCharacterCount

13.2.1.601 SerialTransmitQueueCurrentCharacterCount

`quickSpinIntegerNode` SerialTransmitQueueCurrentCharacterCount

13.2.1.602 SerialTransmitQueueMaxCharacterCount

`quickSpinIntegerNode` SerialTransmitQueueMaxCharacterCount

13.2.1.603 Sharpening

`quickSpinFloatNode` Sharpening

13.2.1.604 SharpeningAuto

`quickSpinBooleanNode` SharpeningAuto

13.2.1.605 SharpeningEnable

`quickSpinBooleanNode` SharpeningEnable

13.2.1.606 SharpeningThreshold

`quickSpinFloatNode` SharpeningThreshold

13.2.1.607 SoftwareSignalPulse

`quickSpinCommandNode` SoftwareSignalPulse

13.2.1.608 SoftwareSignalSelector

`quickSpinEnumerationNode` SoftwareSignalSelector

13.2.1.609 SourceCount

`quickSpinIntegerNode` SourceCount

13.2.1.610 SourceSelector

`quickSpinEnumerationNode` SourceSelector

13.2.1.611 Test0001

`quickSpinIntegerNode` Test0001

13.2.1.612 TestEventGenerate

`quickSpinCommandNode` TestEventGenerate

13.2.1.613 TestPattern

`quickSpinEnumerationNode` TestPattern

13.2.1.614 TestPatternGeneratorSelector

`quickSpinEnumerationNode` TestPatternGeneratorSelector

13.2.1.615 TestPendingAck

`quickSpinIntegerNode` TestPendingAck

13.2.1.616 TimerDelay

`quickSpinFloatNode` TimerDelay

13.2.1.617 TimerDuration

`quickSpinFloatNode` TimerDuration

13.2.1.618 TimerReset

`quickSpinCommandNode` TimerReset

13.2.1.619 TimerSelector

`quickSpinEnumerationNode` TimerSelector

13.2.1.620 TimerStatus

`quickSpinEnumerationNode` TimerStatus

13.2.1.621 TimerTriggerActivation

[quickSpinEnumerationNode](#) TimerTriggerActivation

13.2.1.622 TimerTriggerSource

[quickSpinEnumerationNode](#) TimerTriggerSource

13.2.1.623 TimerValue

[quickSpinFloatNode](#) TimerValue

13.2.1.624 Timestamp

[quickSpinIntegerNode](#) Timestamp

13.2.1.625 TimestampLatch

[quickSpinCommandNode](#) TimestampLatch

13.2.1.626 TimestampLatchValue

[quickSpinIntegerNode](#) TimestampLatchValue

13.2.1.627 TimestampReset

[quickSpinCommandNode](#) TimestampReset

13.2.1.628 TLParamsLocked

[quickSpinIntegerNode](#) TLParamsLocked

13.2.1.629 TransferAbort

[quickSpinCommandNode](#) TransferAbort

13.2.1.630 TransferBlockCount

[quickSpinIntegerNode](#) TransferBlockCount

13.2.1.631 TransferBurstCount

[quickSpinIntegerNode](#) TransferBurstCount

13.2.1.632 TransferComponentSelector

[quickSpinEnumerationNode](#) TransferComponentSelector

13.2.1.633 TransferControlMode

[quickSpinEnumerationNode](#) TransferControlMode

13.2.1.634 TransferOperationMode

[quickSpinEnumerationNode](#) TransferOperationMode

13.2.1.635 TransferPause

[quickSpinCommandNode](#) TransferPause

13.2.1.636 TransferQueueCurrentBlockCount

[quickSpinIntegerNode](#) TransferQueueCurrentBlockCount

13.2.1.637 TransferQueueMaxBlockCount

[quickSpinIntegerNode](#) TransferQueueMaxBlockCount

13.2.1.638 TransferQueueMode

[quickSpinEnumerationNode](#) TransferQueueMode

13.2.1.639 TransferQueueOverflowCount

[quickSpinIntegerNode](#) TransferQueueOverflowCount

13.2.1.640 TransferResume

[quickSpinCommandNode](#) TransferResume

13.2.1.641 TransferSelector

[quickSpinEnumerationNode](#) TransferSelector

13.2.1.642 TransferStart

[quickSpinCommandNode](#) TransferStart

13.2.1.643 TransferStatus

[quickSpinBooleanNode](#) TransferStatus

13.2.1.644 TransferStatusSelector

[quickSpinEnumerationNode](#) TransferStatusSelector

13.2.1.645 TransferStop

`quickSpinCommandNode` TransferStop

13.2.1.646 TransferStreamChannel

`quickSpinIntegerNode` TransferStreamChannel

13.2.1.647 TransferTriggerActivation

`quickSpinEnumerationNode` TransferTriggerActivation

13.2.1.648 TransferTriggerMode

`quickSpinEnumerationNode` TransferTriggerMode

13.2.1.649 TransferTriggerSelector

`quickSpinEnumerationNode` TransferTriggerSelector

13.2.1.650 TransferTriggerSource

`quickSpinEnumerationNode` TransferTriggerSource

13.2.1.651 TriggerActivation

`quickSpinEnumerationNode` TriggerActivation

13.2.1.652 TriggerDelay

`quickSpinFloatNode` TriggerDelay

13.2.1.653 TriggerDivider

`quickSpinIntegerNode` TriggerDivider

13.2.1.654 TriggerEventTest

`quickSpinCommandNode` TriggerEventTest

13.2.1.655 TriggerMode

`quickSpinEnumerationNode` TriggerMode

13.2.1.656 TriggerMultiplier

`quickSpinIntegerNode` TriggerMultiplier

13.2.1.657 TriggerOverlap

`quickSpinEnumerationNode` TriggerOverlap

13.2.1.658 TriggerSelector

`quickSpinEnumerationNode` TriggerSelector

13.2.1.659 TriggerSoftware

`quickSpinCommandNode` TriggerSoftware

13.2.1.660 TriggerSource

`quickSpinEnumerationNode` TriggerSource

13.2.1.661 UserOutputSelector

`quickSpinEnumerationNode` UserOutputSelector

13.2.1.662 UserOutputValue

`quickSpinBooleanNode` UserOutputValue

13.2.1.663 UserOutputValueAll

`quickSpinIntegerNode` UserOutputValueAll

13.2.1.664 UserOutputValueAllMask

`quickSpinIntegerNode` UserOutputValueAllMask

13.2.1.665 UserSetDefault

`quickSpinEnumerationNode` UserSetDefault

13.2.1.666 UserSetFeatureEnable

`quickSpinBooleanNode` UserSetFeatureEnable

13.2.1.667 UserSetLoad

`quickSpinCommandNode` UserSetLoad

13.2.1.668 UserSetSave

`quickSpinCommandNode` UserSetSave

13.2.1.669 UserSetSelector

[quickSpinEnumerationNode](#) UserSetSelector

13.2.1.670 V3_3Enable

[quickSpinBooleanNode](#) V3_3Enable

13.2.1.671 WhiteClip

[quickSpinFloatNode](#) WhiteClip

13.2.1.672 WhiteClipSelector

[quickSpinEnumerationNode](#) WhiteClipSelector

13.2.1.673 Width

[quickSpinIntegerNode](#) Width

13.2.1.674 WidthMax

[quickSpinIntegerNode](#) WidthMax

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

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

13.3 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

13.3.1 Field Documentation

13.3.1.1 DeviceAccessStatus

[quickSpinEnumerationNode](#) DeviceAccessStatus

13.3.1.2 DeviceBootloaderVersion

[quickSpinIntegerNode](#) DeviceBootloaderVersion

13.3.1.3 DeviceCurrentSpeed

[quickSpinEnumerationNode](#) DeviceCurrentSpeed

13.3.1.4 DeviceDisplayName

[quickSpinStringNode](#) DeviceDisplayName

13.3.1.5 DeviceDriverVersion

[quickSpinStringNode](#) DeviceDriverVersion

13.3.1.6 DeviceEndiannessMechanism

[quickSpinEnumerationNode](#) DeviceEndiannessMechanism

13.3.1.7 DeviceID

[quickSpinStringNode](#) DeviceID

13.3.1.8 DeviceInstanceId

[quickSpinStringNode](#) DeviceInstanceId

13.3.1.9 DeviceIsUpdater

`quickSpinBooleanNode` DeviceIsUpdater

13.3.1.10 DeviceLinkSpeed

`quickSpinIntegerNode` DeviceLinkSpeed

13.3.1.11 DeviceLocation

`quickSpinStringNode` DeviceLocation

13.3.1.12 DeviceModelName

`quickSpinStringNode` DeviceModelName

13.3.1.13 DeviceMulticastMonitorMode

`quickSpinBooleanNode` DeviceMulticastMonitorMode

13.3.1.14 DevicePortId

`quickSpinStringNode` DevicePortId

13.3.1.15 DeviceReset

`quickSpinCommandNode` DeviceReset

13.3.1.16 DeviceSerialNumber

`quickSpinStringNode` DeviceSerialNumber

13.3.1.17 DeviceType

[quickSpinEnumerationNode](#) DeviceType

13.3.1.18 DeviceU3VProtocol

[quickSpinBooleanNode](#) DeviceU3VProtocol

13.3.1.19 DeviceUserID

[quickSpinStringNode](#) DeviceUserID

13.3.1.20 DeviceVendorName

[quickSpinStringNode](#) DeviceVendorName

13.3.1.21 DeviceVersion

[quickSpinStringNode](#) DeviceVersion

13.3.1.22 GenICamXMLLocation

[quickSpinEnumerationNode](#) GenICamXMLLocation

13.3.1.23 GenICamXMLPath

[quickSpinStringNode](#) GenICamXMLPath

13.3.1.24 GevCCP

[quickSpinEnumerationNode](#) GevCCP

13.3.1.25 **GevDeviceAutoForceIP**

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

13.3.1.26 **GevDeviceDiscoverMaximumPacketSize**

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

13.3.1.27 **GevDeviceForceGateway**

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

13.3.1.28 **GevDeviceForceIP**

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

13.3.1.29 **GevDeviceForceIPAddress**

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

13.3.1.30 **GevDeviceForceSubnetMask**

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

13.3.1.31 **GevDeviceGateway**

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

13.3.1.32 **GevDeviceIPAddress**

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

13.3.1.33 **GevDeviceIsWrongSubnet**

`quickSpinBooleanNode` `GevDeviceIsWrongSubnet`

13.3.1.34 **GevDeviceMACAddress**

`quickSpinIntegerNode` `GevDeviceMACAddress`

13.3.1.35 **GevDeviceMaximumPacketSize**

`quickSpinIntegerNode` `GevDeviceMaximumPacketSize`

13.3.1.36 **GevDeviceMaximumRetryCount**

`quickSpinIntegerNode` `GevDeviceMaximumRetryCount`

13.3.1.37 **GevDeviceModelsBigEndian**

`quickSpinBooleanNode` `GevDeviceModeIsBigEndian`

13.3.1.38 **GevDevicePort**

`quickSpinIntegerNode` `GevDevicePort`

13.3.1.39 **GevDeviceReadAndWriteTimeout**

`quickSpinIntegerNode` `GevDeviceReadAndWriteTimeout`

13.3.1.40 **GevDeviceSubnetMask**

`quickSpinIntegerNode` `GevDeviceSubnetMask`

13.3.1.41 **GevVersionMajor**

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

13.3.1.42 **GevVersionMinor**

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

13.3.1.43 **GUIXMLLocation**

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

13.3.1.44 **GUIXMLPath**

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

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

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

13.4 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](#)
- [quickSpinBooleanNode](#) [GevInterfaceIsIPConflict](#)
- [quickSpinEnumerationNode](#) [POEStatus](#)
- [quickSpinEnumerationNode](#) [FLIRFilterDriverStatus](#)
- [quickSpinEnumerationNode](#) [TeledyneGigeVisionFilterDriverStatus](#)
- [quickSpinIntegerNode](#) [GevActionDeviceKey](#)
- [quickSpinIntegerNode](#) [GevActionGroupKey](#)

- [quickSpinIntegerNode](#) [GevActionGroupMask](#)
- [quickSpinIntegerNode](#) [GevActionTime](#)
- [quickSpinBooleanNode](#) [GevActionAckRequired](#)
- [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](#)
- [quickSpinBooleanNode](#) [GevDeviceDiscoveryEnabled](#)
- [quickSpinCommandNode](#) [GevDeviceEnableDiscovery](#)
- [quickSpinCommandNode](#) [GevDeviceDisableDiscovery](#)
- [quickSpinStringNode](#) [HostAdapterName](#)
- [quickSpinStringNode](#) [HostAdapterVendor](#)
- [quickSpinStringNode](#) [HostAdapterDriverVersion](#)

13.4.1 Field Documentation

13.4.1.1 ActionCommand

[quickSpinCommandNode](#) [ActionCommand](#)

13.4.1.2 DeviceAccessStatus

[quickSpinEnumerationNode](#) [DeviceAccessStatus](#)

13.4.1.3 DeviceCount

`quickSpinIntegerNode` DeviceCount

13.4.1.4 DeviceID

`quickSpinStringNode` DeviceID

13.4.1.5 DeviceModelName

`quickSpinStringNode` DeviceModelName

13.4.1.6 DeviceSelector

`quickSpinIntegerNode` DeviceSelector

13.4.1.7 DeviceSerialNumber

`quickSpinStringNode` DeviceSerialNumber

13.4.1.8 DeviceUnlock

`quickSpinStringNode` DeviceUnlock

13.4.1.9 DeviceUpdateList

`quickSpinCommandNode` DeviceUpdateList

13.4.1.10 DeviceVendorName

`quickSpinStringNode` DeviceVendorName

13.4.1.11 FLIRFilterDriverStatus

`quickSpinEnumerationNode` FLIRFilterDriverStatus

13.4.1.12 GevActionAckRequired

`quickSpinBooleanNode` GevActionAckRequired

13.4.1.13 GevActionDeviceKey

`quickSpinIntegerNode` GevActionDeviceKey

13.4.1.14 GevActionGroupKey

`quickSpinIntegerNode` GevActionGroupKey

13.4.1.15 GevActionGroupMask

`quickSpinIntegerNode` GevActionGroupMask

13.4.1.16 GevActionTime

`quickSpinIntegerNode` GevActionTime

13.4.1.17 GevDeviceAutoForceIP

`quickSpinCommandNode` GevDeviceAutoForceIP

13.4.1.18 GevDeviceDisableDiscovery

`quickSpinCommandNode` GevDeviceDisableDiscovery

13.4.1.19 **GevDeviceDiscoveryEnabled**

`quickSpinBooleanNode` `GevDeviceDiscoveryEnabled`

13.4.1.20 **GevDeviceEnableDiscovery**

`quickSpinCommandNode` `GevDeviceEnableDiscovery`

13.4.1.21 **GevDeviceForceGateway**

`quickSpinIntegerNode` `GevDeviceForceGateway`

13.4.1.22 **GevDeviceForceIP**

`quickSpinCommandNode` `GevDeviceForceIP`

13.4.1.23 **GevDeviceForceIPAddress**

`quickSpinIntegerNode` `GevDeviceForceIPAddress`

13.4.1.24 **GevDeviceForceSubnetMask**

`quickSpinIntegerNode` `GevDeviceForceSubnetMask`

13.4.1.25 **GevDeviceGateway**

`quickSpinIntegerNode` `GevDeviceGateway`

13.4.1.26 **GevDeviceIPAddress**

`quickSpinIntegerNode` `GevDeviceIPAddress`

13.4.1.27 **GevDeviceMACAddress**

`quickSpinIntegerNode` `GevDeviceMACAddress`

13.4.1.28 **GevDeviceSubnetMask**

`quickSpinIntegerNode` `GevDeviceSubnetMask`

13.4.1.29 **GevInterfaceGateway**

`quickSpinIntegerNode` `GevInterfaceGateway`

13.4.1.30 **GevInterfaceGatewaySelector**

`quickSpinIntegerNode` `GevInterfaceGatewaySelector`

13.4.1.31 **GevInterfaceIsIPConflict**

`quickSpinBooleanNode` `GevInterfaceIsIPConflict`

13.4.1.32 **GevInterfaceMACAddress**

`quickSpinIntegerNode` `GevInterfaceMACAddress`

13.4.1.33 **GevInterfaceMTU**

`quickSpinIntegerNode` `GevInterfaceMTU`

13.4.1.34 **GevInterfaceReceiveLinkSpeed**

`quickSpinIntegerNode` `GevInterfaceReceiveLinkSpeed`

13.4.1.35 GevInterfaceSubnetIPAddress

`quickSpinIntegerNode` GevInterfaceSubnetIPAddress

13.4.1.36 GevInterfaceSubnetMask

`quickSpinIntegerNode` GevInterfaceSubnetMask

13.4.1.37 GevInterfaceSubnetSelector

`quickSpinIntegerNode` GevInterfaceSubnetSelector

13.4.1.38 GevInterfaceTransmitLinkSpeed

`quickSpinIntegerNode` GevInterfaceTransmitLinkSpeed

13.4.1.39 HostAdapterDriverVersion

`quickSpinStringNode` HostAdapterDriverVersion

13.4.1.40 HostAdapterName

`quickSpinStringNode` HostAdapterName

13.4.1.41 HostAdapterVendor

`quickSpinStringNode` HostAdapterVendor

13.4.1.42 IncompatibleDeviceCount

`quickSpinIntegerNode` IncompatibleDeviceCount

13.4.1.43 IncompatibleDeviceID

`quickSpinStringNode` IncompatibleDeviceID

13.4.1.44 IncompatibleDeviceModelName

`quickSpinStringNode` IncompatibleDeviceModelName

13.4.1.45 IncompatibleDeviceSelector

`quickSpinIntegerNode` IncompatibleDeviceSelector

13.4.1.46 IncompatibleDeviceVendorName

`quickSpinStringNode` IncompatibleDeviceVendorName

13.4.1.47 IncompatibleGevDeviceIPAddress

`quickSpinIntegerNode` IncompatibleGevDeviceIPAddress

13.4.1.48 IncompatibleGevDeviceMACAddress

`quickSpinIntegerNode` IncompatibleGevDeviceMACAddress

13.4.1.49 IncompatibleGevDeviceSubnetMask

`quickSpinIntegerNode` IncompatibleGevDeviceSubnetMask

13.4.1.50 InterfaceDisplayName

`quickSpinStringNode` InterfaceDisplayName

13.4.1.51 InterfaceID

[quickSpinStringNode](#) [InterfaceID](#)

13.4.1.52 InterfaceType

[quickSpinEnumerationNode](#) [InterfaceType](#)

13.4.1.53 POEStatus

[quickSpinEnumerationNode](#) [POEStatus](#)

13.4.1.54 TeledyneGigeVisionFilterDriverStatus

[quickSpinEnumerationNode](#) [TeledyneGigeVisionFilterDriverStatus](#)

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

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

13.5 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 StreamPacketResendRequestTimeoutCount](#)
- [quickSpinIntegerNode StreamPacketResendRequestedPacketCount](#)
- [quickSpinIntegerNode StreamPacketResendReceivedPacketCount](#)
- [quickSpinIntegerNode StreamPacketsDuplicatedCount](#)
- [quickSpinIntegerNode StreamPacketsTimeoutCount](#)
- [quickSpinIntegerNode StreamPacketsNotYetAvailableCount](#)
- [quickSpinIntegerNode StreamPacketsTemporarilyUnavailableCount](#)
- [quickSpinIntegerNode StreamPacketsPerFrameCount](#)
- [quickSpinIntegerNode StreamPacketsUnavailableCount](#)
- [quickSpinIntegerNode StreamBlocksReceptionTimeLast](#)
- [quickSpinIntegerNode StreamBlocksReceptionTimeMin](#)
- [quickSpinIntegerNode StreamBlocksReceptionTimeMax](#)
- [quickSpinIntegerNode StreamBlocksProcessingTimeLast](#)
- [quickSpinIntegerNode StreamBlocksProcessingTimeMin](#)
- [quickSpinIntegerNode StreamBlocksProcessingTimeMax](#)
- [quickSpinIntegerNode StreamBlockTransferSize](#)

13.5.1 Field Documentation

13.5.1.1 StreamAnnounceBufferMinimum

[quickSpinIntegerNode](#) `StreamAnnounceBufferMinimum`

13.5.1.2 StreamAnnouncedBufferCount

[quickSpinIntegerNode](#) `StreamAnnouncedBufferCount`

13.5.1.3 StreamBlocksProcessingTimeLast

[quickSpinIntegerNode](#) `StreamBlocksProcessingTimeLast`

13.5.1.4 StreamBlocksProcessingTimeMax

`quickSpinIntegerNode` StreamBlocksProcessingTimeMax

13.5.1.5 StreamBlocksProcessingTimeMin

`quickSpinIntegerNode` StreamBlocksProcessingTimeMin

13.5.1.6 StreamBlocksReceptionTimeLast

`quickSpinIntegerNode` StreamBlocksReceptionTimeLast

13.5.1.7 StreamBlocksReceptionTimeMax

`quickSpinIntegerNode` StreamBlocksReceptionTimeMax

13.5.1.8 StreamBlocksReceptionTimeMin

`quickSpinIntegerNode` StreamBlocksReceptionTimeMin

13.5.1.9 StreamBlockTransferSize

`quickSpinIntegerNode` StreamBlockTransferSize

13.5.1.10 StreamBufferAlignment

`quickSpinIntegerNode` StreamBufferAlignment

13.5.1.11 StreamBufferCountManual

`quickSpinIntegerNode` StreamBufferCountManual

13.5.1.12 StreamBufferCountMax

`quickSpinIntegerNode` `StreamBufferCountMax`

13.5.1.13 StreamBufferCountMode

`quickSpinEnumerationNode` `StreamBufferCountMode`

13.5.1.14 StreamBufferCountResult

`quickSpinIntegerNode` `StreamBufferCountResult`

13.5.1.15 StreamBufferHandlingMode

`quickSpinEnumerationNode` `StreamBufferHandlingMode`

13.5.1.16 StreamChunkCountMaximum

`quickSpinIntegerNode` `StreamChunkCountMaximum`

13.5.1.17 StreamCRCCheckEnable

`quickSpinBooleanNode` `StreamCRCCheckEnable`

13.5.1.18 StreamDeliveredFrameCount

`quickSpinIntegerNode` `StreamDeliveredFrameCount`

13.5.1.19 StreamDroppedFrameCount

`quickSpinIntegerNode` `StreamDroppedFrameCount`

13.5.1.20 StreamID

`quickSpinStringNode` StreamID

13.5.1.21 StreamIncompleteFrameCount

`quickSpinIntegerNode` StreamIncompleteFrameCount

13.5.1.22 StreamInputBufferCount

`quickSpinIntegerNode` StreamInputBufferCount

13.5.1.23 StreamIsGrabbing

`quickSpinBooleanNode` StreamIsGrabbing

13.5.1.24 StreamLostFrameCount

`quickSpinIntegerNode` StreamLostFrameCount

13.5.1.25 StreamMissedPacketCount

`quickSpinIntegerNode` StreamMissedPacketCount

13.5.1.26 StreamMode

`quickSpinEnumerationNode` StreamMode

13.5.1.27 StreamOutputBufferCount

`quickSpinIntegerNode` StreamOutputBufferCount

13.5.1.28 StreamPacketResendEnable

`quickSpinBooleanNode` `StreamPacketResendEnable`

13.5.1.29 StreamPacketResendMaxRequests

`quickSpinIntegerNode` `StreamPacketResendMaxRequests`

13.5.1.30 StreamPacketResendReceivedPacketCount

`quickSpinIntegerNode` `StreamPacketResendReceivedPacketCount`

13.5.1.31 StreamPacketResendRequestCount

`quickSpinIntegerNode` `StreamPacketResendRequestCount`

13.5.1.32 StreamPacketResendRequestedPacketCount

`quickSpinIntegerNode` `StreamPacketResendRequestedPacketCount`

13.5.1.33 StreamPacketResendRequestTimeoutCount

`quickSpinIntegerNode` `StreamPacketResendRequestTimeoutCount`

13.5.1.34 StreamPacketResendTimeout

`quickSpinIntegerNode` `StreamPacketResendTimeout`

13.5.1.35 StreamPacketsDuplicatedCount

`quickSpinIntegerNode` `StreamPacketsDuplicatedCount`

13.5.1.36 StreamPacketsNotYetAvailableCount

`quickSpinIntegerNode` StreamPacketsNotYetAvailableCount

13.5.1.37 StreamPacketsPerFrameCount

`quickSpinIntegerNode` StreamPacketsPerFrameCount

13.5.1.38 StreamPacketsTemporarilyUnavailableCount

`quickSpinIntegerNode` StreamPacketsTemporarilyUnavailableCount

13.5.1.39 StreamPacketsTimeoutCount

`quickSpinIntegerNode` StreamPacketsTimeoutCount

13.5.1.40 StreamPacketsUnavailableCount

`quickSpinIntegerNode` StreamPacketsUnavailableCount

13.5.1.41 StreamReceivedFrameCount

`quickSpinIntegerNode` StreamReceivedFrameCount

13.5.1.42 StreamReceivedPacketCount

`quickSpinIntegerNode` StreamReceivedPacketCount

13.5.1.43 StreamStartedFrameCount

`quickSpinIntegerNode` StreamStartedFrameCount

13.5.1.44 StreamType

[quickSpinEnumerationNode](#) StreamType

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

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

13.6 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
- [quickSpinBooleanNode](#) GevAutoAssignIPEnable

13.6.1 Field Documentation

13.6.1.1 EnumerateGen2Cameras

[quickSpinBooleanNode](#) EnumerateGen2Cameras

13.6.1.2 EnumerateGEVInterfaces

`quickSpinBooleanNode` EnumerateGEVInterfaces

13.6.1.3 EnumerateUSBInterfaces

`quickSpinBooleanNode` EnumerateUSBInterfaces

13.6.1.4 GenTLFNCVersionMajor

`quickSpinIntegerNode` GenTLFNCVersionMajor

13.6.1.5 GenTLFNCVersionMinor

`quickSpinIntegerNode` GenTLFNCVersionMinor

13.6.1.6 GenTLFNCVersionSubMinor

`quickSpinIntegerNode` GenTLFNCVersionSubMinor

13.6.1.7 GenTLVersionMajor

`quickSpinIntegerNode` GenTLVersionMajor

13.6.1.8 GenTLVersionMinor

`quickSpinIntegerNode` GenTLVersionMinor

13.6.1.9 GevAutoAssignIPEnable

`quickSpinBooleanNode` GevAutoAssignIPEnable

13.6.1.10 **GevInterfaceDefaultGateway**

`quickSpinIntegerNode` `GevInterfaceDefaultGateway`

13.6.1.11 **GevInterfaceDefaultIPAddress**

`quickSpinIntegerNode` `GevInterfaceDefaultIPAddress`

13.6.1.12 **GevInterfaceDefaultSubnetMask**

`quickSpinIntegerNode` `GevInterfaceDefaultSubnetMask`

13.6.1.13 **GevInterfaceMACAddress**

`quickSpinIntegerNode` `GevInterfaceMACAddress`

13.6.1.14 **GevVersionMajor**

`quickSpinIntegerNode` `GevVersionMajor`

13.6.1.15 **GevVersionMinor**

`quickSpinIntegerNode` `GevVersionMinor`

13.6.1.16 **InterfaceDisplayName**

`quickSpinStringNode` `InterfaceDisplayName`

13.6.1.17 **InterfaceID**

`quickSpinStringNode` `InterfaceID`

13.6.1.18 InterfaceSelector

`quickSpinIntegerNode` InterfaceSelector

13.6.1.19 InterfaceUpdateList

`quickSpinCommandNode` InterfaceUpdateList

13.6.1.20 TLDisplayName

`quickSpinStringNode` TLDisplayName

13.6.1.21 TLFileName

`quickSpinStringNode` TLFileName

13.6.1.22 TLID

`quickSpinStringNode` TLID

13.6.1.23 TLModelName

`quickSpinStringNode` TLModelName

13.6.1.24 TLPath

`quickSpinStringNode` TLPath

13.6.1.25 TLType

`quickSpinEnumerationNode` TLType

13.6.1.26 TLVendorName

`quickSpinStringNode` TLVendorName

13.6.1.27 TLVersion

`quickSpinStringNode` TLVersion

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

- include/spinc/TransportLayerSystemC.h

13.7 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]

13.7.1 Detailed Description

Options for saving uncompressed videos.

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

13.7.2 Field Documentation

13.7.2.1 frameRate

`float frameRate`

Frame rate of the stream.

13.7.2.2 height

```
unsigned int height
```

Height of source image.

13.7.2.3 reserved

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

13.7.2.4 width

```
unsigned int width
```

Width of source image.

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

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

13.8 spinBMPOption Struct Reference

Options for saving BMP images.

Data Fields

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

Reserved for future use.

13.8.1 Detailed Description

Options for saving BMP images.

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

13.8.2 Field Documentation

13.8.2.1 indexedColor_8bit

```
bool8_t indexedColor_8bit
```

13.8.2.2 reserved

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

Reserved for future use.

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

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

13.9 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](#)

13.9.1 Detailed Description

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

13.9.2 Field Documentation

13.9.2.1 `m_blackLevel`

```
double m_blackLevel
```

13.9.2.2 `m_compressionMode`

```
int64_t m_compressionMode
```

13.9.2.3 `m_compressionRatio`

```
double m_compressionRatio
```

13.9.2.4 `m_counterValue`

```
int64_t m_counterValue
```

13.9.2.5 m_cRC

```
int64_t m_cRC
```

13.9.2.6 m_encoderValue

```
int64_t m_encoderValue
```

13.9.2.7 m_exposureEndLineStatusAll

```
int64_t m_exposureEndLineStatusAll
```

13.9.2.8 m_exposureTime

```
double m_exposureTime
```

13.9.2.9 m_frameID

```
int64_t m_frameID
```

13.9.2.10 m_gain

```
double m_gain
```

13.9.2.11 m_height

```
int64_t m_height
```

13.9.2.12 m_image

```
int64_t m_image
```

13.9.2.13 m_inferenceConfidence

```
double m_inferenceConfidence
```

13.9.2.14 m_inferenceFrameId

```
int64_t m_inferenceFrameId
```

13.9.2.15 m_inferenceResult

```
int64_t m_inferenceResult
```

13.9.2.16 m_linePitch

```
int64_t m_linePitch
```

13.9.2.17 m_lineStatusAll

```
int64_t m_lineStatusAll
```

13.9.2.18 m_offsetX

```
int64_t m_offsetX
```

13.9.2.19 m_offsetY

```
int64_t m_offsetY
```

13.9.2.20 m_partSelector

```
int64_t m_partSelector
```

13.9.2.21 m_pixelDynamicRangeMax

```
int64_t m_pixelDynamicRangeMax
```

13.9.2.22 m_pixelDynamicRangeMin

```
int64_t m_pixelDynamicRangeMin
```

13.9.2.23 m_scan3dAxisMax

```
double m_scan3dAxisMax
```

13.9.2.24 m_scan3dAxisMin

```
double m_scan3dAxisMin
```

13.9.2.25 m_scan3dCoordinateOffset

```
double m_scan3dCoordinateOffset
```

13.9.2.26 m_scan3dCoordinateReferenceValue

```
double m_scan3dCoordinateReferenceValue
```

13.9.2.27 m_scan3dCoordinateScale

```
double m_scan3dCoordinateScale
```

13.9.2.28 m_scan3dInvalidDataValue

```
double m_scan3dInvalidDataValue
```

13.9.2.29 m_scan3dTransformValue

```
double m_scan3dTransformValue
```

13.9.2.30 m_scanLineSelector

```
int64_t m_scanLineSelector
```

13.9.2.31 m_sequencerSetActive

```
int64_t m_sequencerSetActive
```

13.9.2.32 m_serialDataLength

```
int64_t m_serialDataLength
```

13.9.2.33 m_streamChannelID

```
int64_t m_streamChannelID
```

13.9.2.34 m_timerValue

```
double m_timerValue
```

13.9.2.35 m_timestamp

```
int64_t m_timestamp
```

13.9.2.36 m_timestampLatchValue

```
int64_t m_timestampLatchValue
```

13.9.2.37 m_transferBlockID

```
int64_t m_transferBlockID
```

13.9.2.38 m_transferQueueCurrentBlockCount

```
int64_t m_transferQueueCurrentBlockCount
```

13.9.2.39 m_width

```
int64_t m_width
```

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

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

13.10 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.

13.10.1 Detailed Description

Options for saving H264 videos.

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

13.10.2 Field Documentation

13.10.2.1 bitrate

```
unsigned int bitrate
```

Bitrate to encode at.

13.10.2.2 frameRate

```
float frameRate
```

Frame rate of the stream.

13.10.2.3 height

```
unsigned int height
```

Height of source image.

13.10.2.4 reserved

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

Reserved for future use.

13.10.2.5 width

```
unsigned int width
```

Width of source image.

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

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

13.11 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.

13.11.1 Detailed Description

Options for saving JPEG images.

Used in saving PPM images with a call to `spinImageSaveJpeg()`.

13.11.2 Field Documentation

13.11.2.1 `progressive`

`bool8_t progressive`

Whether to save as a progressive JPEG file.

13.11.2.2 `quality`

`unsigned int quality`

JPEG image quality in range (0-100).

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

13.11.2.3 `reserved`

`unsigned int reserved[16]`

Reserved for future use.

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

- `include/spinc/SpinnakerDefsC.h`

13.12 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.

13.12.1 Detailed Description

Options for saving JPEG 2000 images.

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

13.12.2 Field Documentation

13.12.2.1 quality

```
unsigned int quality
```

JPEG saving quality in range (1-512).

13.12.2.2 reserved

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

Reserved for future use.

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

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

13.13 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.

13.13.1 Detailed Description

Provides easier access to the current version of Spinnaker.

13.13.2 Field Documentation

13.13.2.1 build

```
unsigned int build
```

Build number of the library.

13.13.2.2 major

```
unsigned int major
```

Major version of the library.

13.13.2.3 minor

```
unsigned int minor
```

Minor version of the library.

13.13.2.4 type

`unsigned int type`

Version type of the library.

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

- `include/spinc/SpinnakerDefsC.h`

13.14 spinMJPEGOption 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]

13.14.1 Detailed Description

Options for saving MJPG videos.

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

13.14.2 Field Documentation

13.14.2.1 frameRate

`float frameRate`

Frame rate of the stream.

13.14.2.2 height

```
unsigned int height
```

Height of source image.

13.14.2.3 quality

```
unsigned int quality
```

Image quality (1-100)

13.14.2.4 reserved

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

13.14.2.5 width

```
unsigned int width
```

Width of source image.

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

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

13.15 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.

13.15.1 Detailed Description

Options for saving PGM images.

13.15.2 Field Documentation

13.15.2.1 binaryFile

`bool8_t binaryFile`

Whether to save the PPM as a binary file.

13.15.2.2 reserved

`unsigned int reserved[16]`

Reserved for future use.

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

- `include/spinc/SpinnakerDefsC.h`

13.16 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.

13.16.1 Detailed Description

Options for saving PNG images.

Used in saving PNG images with a call to `spinImageSavePng()`.

13.16.2 Field Documentation

13.16.2.1 compressionLevel

```
unsigned int compressionLevel
```

Compression level (0-9).

0 is no compression, 9 is best compression.

13.16.2.2 interlaced

```
bool8_t interlaced
```

Whether to save the PNG as interlaced.

13.16.2.3 reserved

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

Reserved for future use.

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

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

13.17 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.

13.17.1 Detailed Description

Options for saving PPM images.

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

13.17.2 Field Documentation

13.17.2.1 binaryFile

`bool8_t` binaryFile

Whether to save the PPM as a binary file.

13.17.2.2 reserved

`unsigned int` reserved[16]

Reserved for future use.

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

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

13.18 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.

13.18.1 Detailed Description

Options for saving TIFF images.

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

13.18.2 Field Documentation

13.18.2.1 compression

`spinTIFFCompressionMethod` compression

Compression method to use for encoding TIFF images.

13.18.2.2 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 14

File Documentation

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

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

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

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

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

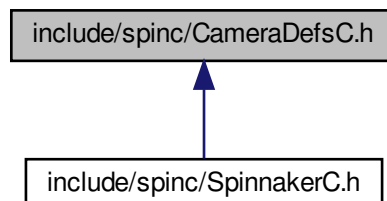
14.6 doc/spindocs/shared/Licensing.dox File Reference

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

14.8 doc/spindocs/shared/NetworkingBestPractices.dox File Reference

14.9 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_DEVICEREGISTERSENDIANNES` }
- 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_DEVICECTLTYPE` }
- 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_DEVICETYPE }
• 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_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_LLCCBayerRG8 ,
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 spinCIConfigurationEnums {

```

```

    ClConfiguration_Base ,
    ClConfiguration_Medium ,
    ClConfiguration_Full ,
    ClConfiguration_DualBase ,
    ClConfiguration_EightyBit ,
    NUM_CLCONFIGURATION }
• enum spinClTimeSlotsCountEnums {
    ClTimeSlotsCount_One ,
    ClTimeSlotsCount_Two ,
    ClTimeSlotsCount_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 ,

```



```

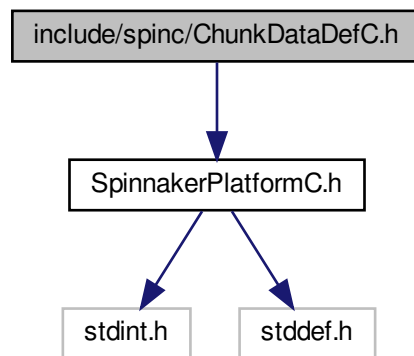
CxpLinkConfigurationPreferred_CXP3_X3 ,
CxpLinkConfigurationPreferred_CXP5_X3 ,
CxpLinkConfigurationPreferred_CXP6_X3 ,
CxpLinkConfigurationPreferred_CXP1_X4 ,
CxpLinkConfigurationPreferred_CXP2_X4 ,
CxpLinkConfigurationPreferred_CXP3_X4 ,
CxpLinkConfigurationPreferred_CXP5_X4 ,
CxpLinkConfigurationPreferred_CXP6_X4 ,
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 ,

```

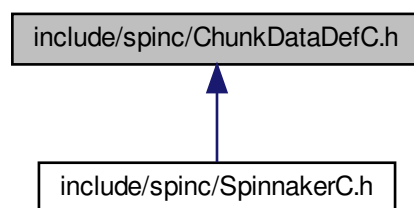
```
CxpPoCxpStatus_Off ,  
CxpPoCxpStatus_Tripped ,  
NUM_CXPPOCXPSTATUS }
```

14.10 include/spinc/ChunkDataDefC.h File Reference

Include dependency graph for ChunkDataDefC.h:



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



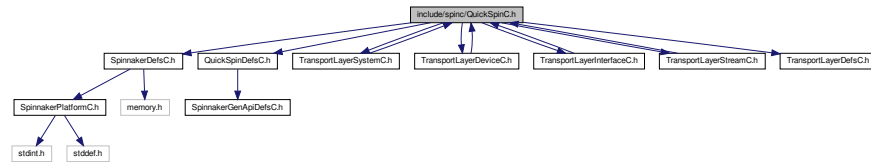
Data Structures

- struct [spinChunkData](#)

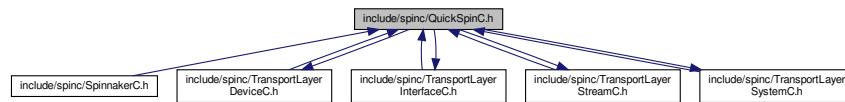
The type of information that can be obtained from image chunk data.

14.11 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](#) *pQuickSpin↔TLDevice)
- [SPINNAKERC_API quickSpinTLStreamInit](#) ([spinCamera](#) hCamera, [quickSpinTLStream](#) *pQuickSpin↔TLStream)
- [SPINNAKERC_API quickSpinTLInterfaceInit](#) ([spinInterface](#) hInterface, [quickSpinTLInterface](#) *pQuickSpin↔TLInterface)
- [SPINNAKERC_API quickSpinTLSystemInit](#) ([spinSystem](#) hSystem, [quickSpinTLSystem](#) *pQuickSpin↔TLSystem)

14.11.1 Function Documentation

14.11.1.1 quickSpinInit()

```

SPINNAKERC_API quickSpinInit (
    spinCamera hCamera,
    quickSpin * pQuickSpin )

```

14.11.1.2 quickSpinInitEx()

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

14.11.1.3 quickSpinTLDeviceInit()

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

14.11.1.4 quickSpinTLInterfaceInit()

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

14.11.1.5 quickSpinTLStreamInit()

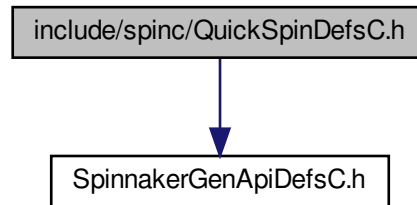
```
SPINNAKERC_API quickSpinTLStreamInit (
    spinCamera hCamera,
    quickSpinTLStream * pQuickSpinTLStream )
```

14.11.1.6 quickSpinTLSystemInit()

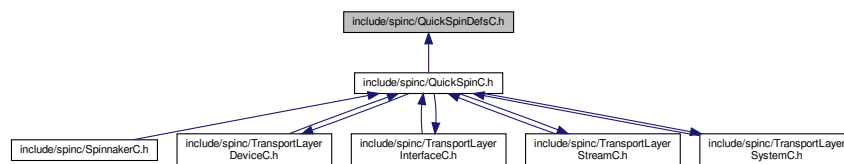
```
SPINNAKERC_API quickSpinTLSystemInit (
    spinSystem hSystem,
    quickSpinTLSystem * pQuickSpinTLSystem )
```

14.12 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](#)

14.12.1 Typedef Documentation

14.12.1.1 quickSpinBooleanNode

```
typedef spinNodeHandle quickSpinBooleanNode
```

14.12.1.2 quickSpinCommandNode

```
typedef spinNodeHandle quickSpinCommandNode
```

14.12.1.3 quickSpinEnumerationNode

```
typedef spinNodeHandle quickSpinEnumerationNode
```

14.12.1.4 quickSpinFloatNode

```
typedef spinNodeHandle quickSpinFloatNode
```

14.12.1.5 quickSpinIntegerNode

```
typedef spinNodeHandle quickSpinIntegerNode
```

14.12.1.6 quickSpinRegisterNode

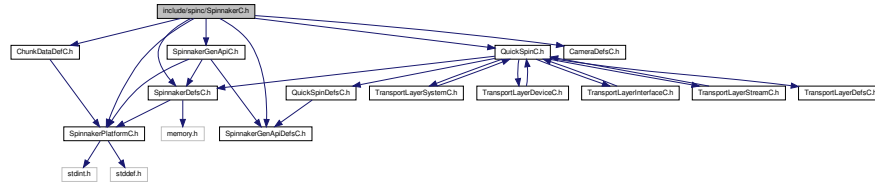
```
typedef spinNodeHandle quickSpinRegisterNode
```

14.12.1.7 quickSpinStringNode

```
typedef spinNodeHandle quickSpinStringNode
```

14.13 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 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, [bool8_t](#) requestAck, [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](#) *phNodeMap)
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, [bool8_t](#) requestAck, [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](#) *phNodeMap)
Retrieves the transport layer device nodemap from a camera.
 - [SPINNAKERC_API spinCameraGetTLStreamNodeMap](#) ([spinCamera](#) hCamera, [spinNodeMapHandle](#) *phNodeMap)
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](#) *phImage)
Retrieves an image from a camera.
- [SPINNAKERC_API spinCameraGetNextImageEx](#) ([spinCamera](#) hCamera, [uint64_t](#) grabTimeout, [spinImage](#) *phImage)
Retrieves an image from a camera; manually set the timeout in milliseconds.
- [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.
- [SPINNAKERC_API spinCameraGetDeviceID](#) ([spinCamera](#) hCamera, char *pBuf, [size_t](#) *pBufLen)
Retrieves a unique identifier for a camera.
- [SPINNAKERC_API_DEPRECATED](#) ("Use [spinCameraGetDeviceID\(\)](#) instead.", [spinCameraGetUniqueID](#)([spinCamera](#) hCamera, char *pBuf, [size_t](#) *pBufLen);) [SPINNAKERC_API spinCamerasStreaming](#)([spinCamera](#) hCamera)
Retrieves a unique identifier for a camera.
- [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 *pMaxPacketSize)

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, [spinPayloadTypeInfolDs](#))
- [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](#) *pIsIncomplete)
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](#) hImage↔Processor, [unsigned int](#) numThreads)

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

Gets the number of threads used for image decompression during [spinImageProcessorConvert\(\)](#).
- [SPINNAKERC_API spinImageProcessorConvert](#) ([spinImageProcessor](#) hImageProcessor, [spinImage](#) hSrc↔Image, [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](#) hImageListEvent↔Handler)
- 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](#) *ph↔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, [spinStatisticsChannel](#) channel, [bool8_t](#) *pbEnabled)
- Checks whether an image statistics context is enabled.*
- [SPINNAKERC_API spinImageStatisticsSetChannelStatus](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#) 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, [spinStatisticsChannel](#) channel, unsigned int *pMin, unsigned int *pMax)

Retrieves the pixel value range of an image statistics channel.

- [SPINNAKERC_API spinImageStatisticsGetNumPixelValues](#) ([spinImageStatistics](#) hStatistics, [spinStatisticsChannel](#) 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)

- [SPINNAKERC_API spinImageChunkDataGetFloatValue](#) ([spinImage](#) hImage, const char *pName, double *pValue)

Variables

- [bool8_t * pblsStreaming](#)

14.13.1 Function Documentation

14.13.1.1 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

14.13.1.2 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

14.13.1.3 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

14.13.1.4 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

14.13.1.5 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

14.13.1.6 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

14.13.1.7 spinCameraGetDeviceID()

```
SPINNAKERC_API spinCameraGetDeviceID (
    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

14.13.1.8 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

14.13.1.9 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

14.13.1.10 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

14.13.1.11 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

14.13.1.12 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

14.13.1.13 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

14.13.1.14 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

14.13.1.15 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

14.13.1.16 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

14.13.1.17 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

14.13.1.18 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

14.13.1.19 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

14.13.1.20 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

14.13.1.21 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

14.13.1.22 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

14.13.1.23 `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

14.13.1.24 `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

14.13.1.25 `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

14.13.1.26 `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

14.13.1.27 spinCameraReadPort()

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

14.13.1.28 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

14.13.1.29 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

14.13.1.30 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

14.13.1.31 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

14.13.1.32 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

14.13.1.33 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

14.13.1.34 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

14.13.1.35 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

14.13.1.36 [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

14.13.1.37 [spinCameraWritePort\(\)](#)

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

14.13.1.38 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: void(spinArrivalEventFunction)(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

14.13.1.39 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

14.13.1.40 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

14.13.1.41 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

14.13.1.42 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

14.13.1.43 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

14.13.1.44 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

14.13.1.45 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

14.13.1.46 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

14.13.1.47 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

14.13.1.48 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

14.13.1.49 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

14.13.1.50 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

14.13.1.51 `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

14.13.1.52 `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

14.13.1.53 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

14.13.1.54 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

14.13.1.55 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

14.13.1.56 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

14.13.1.57 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

14.13.1.58 spinImageChunkDataGetFloatValue()

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

14.13.1.59 spinImageChunkDataGetIntValue()

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

14.13.1.60 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

14.13.1.61 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

14.13.1.62 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

14.13.1.63 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

14.13.1.64 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

14.13.1.65 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

14.13.1.66 `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: <code>void(spinImageEventFunction)(const spinImage hImage, 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

14.13.1.67 `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

14.13.1.68 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

14.13.1.69 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

14.13.1.70 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

14.13.1.71 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

14.13.1.72 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

14.13.1.73 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

14.13.1.74 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

14.13.1.75 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

14.13.1.76 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

14.13.1.77 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

14.13.1.78 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

14.13.1.79 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

14.13.1.80 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

14.13.1.81 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

14.13.1.82 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

14.13.1.83 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

14.13.1.84 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

14.13.1.85 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

14.13.1.86 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

14.13.1.87 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

14.13.1.88 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

14.13.1.89 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

14.13.1.90 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

14.13.1.91 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

14.13.1.92 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

14.13.1.93 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

14.13.1.94 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

14.13.1.95 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

14.13.1.96 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

14.13.1.97 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

14.13.1.98 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

14.13.1.99 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

14.13.1.100 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

14.13.1.101 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

14.13.1.102 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

14.13.1.103 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

14.13.1.104 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

14.13.1.105 [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

14.13.1.106 [spinImageListRelease\(\)](#)

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

14.13.1.107 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

14.13.1.108 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

14.13.1.109 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

14.13.1.110 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) |

14.13.1.111 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. |

14.13.1.112 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. |

14.13.1.113 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

14.13.1.114 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

14.13.1.115 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\(\)](#)

14.13.1.116 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\(\)](#)

14.13.1.117 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\(\)](#)

14.13.1.118 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\(\)](#)**14.13.1.119 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

14.13.1.120 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

14.13.1.121 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

14.13.1.122 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

14.13.1.123 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

14.13.1.124 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

14.13.1.125 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

14.13.1.126 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

14.13.1.127 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

14.13.1.128 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

14.13.1.129 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

14.13.1.130 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

14.13.1.131 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

14.13.1.132 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

14.13.1.133 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

14.13.1.134 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

14.13.1.135 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

14.13.1.136 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

14.13.1.137 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

14.13.1.138 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>ppiHistogram</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

14.13.1.139 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

14.13.1.140 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

14.13.1.141 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

14.13.1.142 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

14.13.1.143 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

14.13.1.144 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

14.13.1.145 spinImageStatisticsSetChannelStatus()

```
SPINNAKER_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

14.13.1.146 spinInterfaceEventHandlerCreate()

```
SPINNAKER_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: <code>void(spinArrivalEventFunction)(void pUserData)</code> |
| <i>hRemovalFunction</i> | The function to be called at removal event occurrences; signature to match: <code>void(spinRemovalEventFunction)(uint64_t deviceSerialNumber, 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

14.13.1.147 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

14.13.1.148 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

14.13.1.149 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

14.13.1.150 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

14.13.1.151 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>pbIsInUse</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

14.13.1.152 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

14.13.1.153 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

14.13.1.154 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

14.13.1.155 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

14.13.1.156 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](#)

14.13.1.157 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

14.13.1.158 `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

14.13.1.159 `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

14.13.1.160 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

14.13.1.161 spinInterfaceSendActionCommand()

```
SPINNAKERC_API spinInterfaceSendActionCommand (  
    spinInterface hInterface,  
    size_t iDeviceKey,  
    size_t iGroupKey,  
    size_t iGroupMask,  
    size_t iActionTime,  
    bool8_t requestAck,  
    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. |
| <i>requestAck</i> | (Optional) Whether to request an ACK from the camera. True is to send ack. |
| <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

14.13.1.162 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

14.13.1.163 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

14.13.1.164 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

14.13.1.165 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

14.13.1.166 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

14.13.1.167 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

14.13.1.168 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

14.13.1.169 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

14.13.1.170 spinLogDataGetPriorityName()

```
SPINNAKER_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

14.13.1.171 spinLogDataGetThreadName()

```
SPINNAKER_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

14.13.1.172 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

14.13.1.173 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

14.13.1.174 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

14.13.1.175 SPINNAKERC_API_DEPRECATED()

```
SPINNAKERC_API_DEPRECATED (
    "Use spinCameraGetDeviceID\(\) instead." ,
    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 Checks whether a camera is currently acquiring images

See also

[spinError](#)

Parameters

| | |
|----------------------|--|
| <i>hCamera</i> | The camera to check |
| <i>pblsStreaming</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

14.13.1.176 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

14.13.1.177 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

14.13.1.178 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

14.13.1.179 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

14.13.1.180 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).

14.13.1.181 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

14.13.1.182 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

14.13.1.183 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

14.13.1.184 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

14.13.1.185 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

14.13.1.186 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

14.13.1.187 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

14.13.1.188 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

14.13.1.189 spinSystemSendActionCommand()

```
SPINNAKERC_API spinSystemSendActionCommand (
    spinSystem hSystem,
```

```

size_t iDeviceKey,
size_t iGroupKey,
size_t iGroupMask,
size_t iActionTime,
bool8_t requestAck,
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>requestAck</i> | (Optional) Whether to request an ACK from the camera. True is to send ack. |
| <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

14.13.1.190 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

14.13.1.191 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

14.13.1.192 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

14.13.1.193 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

14.13.1.194 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

14.13.1.195 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

14.13.1.196 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

14.13.1.197 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

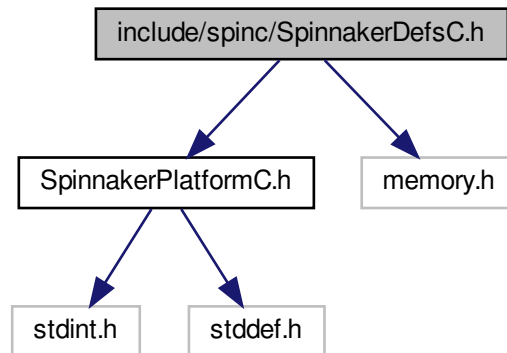
`spinError` The error code; returns `SPINNAKER_ERR_SUCCESS` (or 0) for no error

14.13.2 Variable Documentation**14.13.2.1 pblsStreaming**

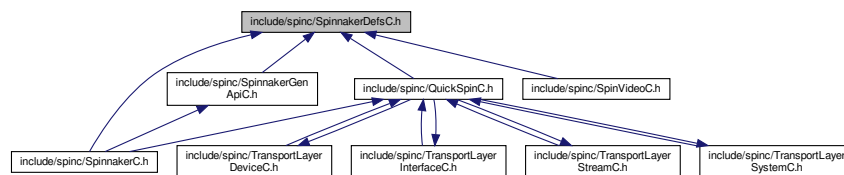
```
bool8_t* pbIsStreaming
```


14.14 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 [spinMJPGOption](#)

- 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_TLPIXELFORMAT_NAMESPACE_UNKNOWN` = 0 ,
`SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV` = 1 ,
`SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC` = 2 ,
`SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT` = 3 ,
`SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT` = 4 ,
`SPINNAKER_PIXELFORMAT_NAMESPACE_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` ,
`SPINNAKER_IMAGE_STATUS_MISSING_TRAILER` ,
`SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE` ,
`SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT` ,
`SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID` = 11 ,
`SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES` = 12 }

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

- enum `spinnakerLogLevel` {
`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_TLPAYLOAD_TYPE_UNKNOWN` = 0 ,
`SPINNAKER_TLPAYLOAD_TYPE_IMAGE` = 1 ,
`SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA` = 2 ,
`SPINNAKER_TLPAYLOAD_TYPE_FILE` = 3 ,
`SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA` = 4 ,
`SPINNAKER_TLPAYLOAD_TYPE_JPEG` = 5 ,
`SPINNAKER_TLPAYLOAD_TYPE_JPEG2000` = 6 ,
`SPINNAKER_TLPAYLOAD_TYPE_H264` = 7 ,
`SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY` = 8 ,
`SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC` = 9 ,
`SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART` = 10 ,
`SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID` = 1000 ,
`SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED` = `SPINNAKER_TLPAYLOAD_TYPE_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

14.14.1 Typedef Documentation

14.14.1.1 bool8_t

```
typedef uint8_t bool8_t
```

14.14.1.2 spinArrivalEventFunction

```
typedef void(* spinArrivalEventFunction) (const spinCamera hCamera, void *pUserData)
```

14.14.1.3 spinCamera

```
typedef void* spinCamera
```

Handle for camera functionality.

Created by calling [spinCameraListGet\(\)](#), which requires a call to [spinCameraRelease\(\)](#) to release.

14.14.1.4 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.

14.14.1.5 spinDeviceArrivalEventHandler

```
typedef void* spinDeviceArrivalEventHandler
```

Handle for arrival event handler functionality.

Created by calling [spinArrivalEventCreate\(\)](#), which requires a call to [spinDeviceArrivalEventHandlerDestroy\(\)](#) to destroy.

14.14.1.6 spinDeviceEventData

```
typedef void* spinDeviceEventData
```

Handle for device event data functionality.

Received in device event function. No need to release, clear, or destroy.

14.14.1.7 spinDeviceEventFunction

```
typedef void(* spinDeviceEventFunction) (const spinDeviceEventData hEventData, const char *p↔  
EventName, void *pUserData)
```

Function signatures are used to create and trigger callbacks and events.

14.14.1.8 spinDeviceEventHandler

```
typedef void* spinDeviceEventHandler
```

Handle for device event handler functionality.

Created by calling [spinDeviceEventHandlerCreate\(\)](#), which requires a call to [spinDeviceEventHandlerDestroy\(\)](#) to destroy.

14.14.1.9 spinDeviceRemovalEventHandler

```
typedef void* spinDeviceRemovalEventHandler
```

Handle for removal event handler functionality.

Created by calling [spinDeviceRemovalEventHandlerCreate\(\)](#), which requires a call to [spinDeviceRemovalEventHandlerDestroy\(\)](#) to destroy.

14.14.1.10 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.

14.14.1.11 spinImageEventFunction

```
typedef void(* spinImageEventFunction) (const spinImage hImage, void *pUserData)
```

14.14.1.12 spinImageEventHandler

```
typedef void* spinImageEventHandler
```

Handle for image event handler functionality.

Created by calling [spinImageEventHandlerCreate\(\)](#), which requires a call to [spinImageEventHandlerDestroy\(\)](#) to destroy.

14.14.1.13 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.

14.14.1.14 spinImageListEventFunction

```
typedef void(* spinImageListEventFunction) (const spinImageList hImage, void *pUserData)
```

14.14.1.15 spinImageListEventHandler

```
typedef void* spinImageListEventHandler
```

Handle for image list event handler functionality.

Created by calling [spinImageListEventHandlerCreate\(\)](#), which requires a call to [spinImageListEventHandlerDestroy\(\)](#) to destroy.

14.14.1.16 spinImageProcessor

```
typedef void* spinImageProcessor
```

Handle for image processor functionality.

Created by calling [spinImageProcessorCreate\(\)](#), which requires a call to [spinImageProcessorDestroy\(\)](#) to destroy.

14.14.1.17 spinImageStatistics

```
typedef void* spinImageStatistics
```

Handle for image statistics functionality.

Created by calling [spinImageStatisticsCreate\(\)](#), which requires a call to [spinImageStatisticsDestroy\(\)](#) to destroy.

14.14.1.18 spinInterface

```
typedef void* spinInterface
```

Handle for interface functionality.

Created by calling [spinInterfaceListGet\(\)](#), which requires a call to [spinInterfaceRelease\(\)](#) to release.

14.14.1.19 spinInterfaceEventHandler

```
typedef void* spinInterfaceEventHandler
```

Handle for interface event handler functionality.

Created by calling [spinInterfaceEventHandlerCreate\(\)](#), which requires a call to [spinInterfaceEventHandlerDestroy\(\)](#) to destroy.

14.14.1.20 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.

14.14.1.21 spinLogEventData

```
typedef void* spinLogEventData
```

Handle for logging event data functionality.

Received in log event function. No need to release, clear, or destroy.

14.14.1.22 spinLogEventFunction

```
typedef void(* spinLogEventFunction) (const spinLogEventData hEventData, void *pUserData)
```

14.14.1.23 spinLogEventHandler

```
typedef void* spinLogEventHandler
```

Handle for logging event handler functionality.

Created by calling [spinLogEventHandlerCreate\(\)](#), which requires a call to [spinLogEventHandlerDestroy\(\)](#) to destroy.

14.14.1.24 spinRemovalEventFunction

```
typedef void(* spinRemovalEventFunction) (const spinCamera hCamera, void *pUserData)
```

14.14.1.25 spinSystem

```
typedef void* spinSystem
```

Handle for system functionality.

Created by calling [spinSystemGetInstance\(\)](#), which requires a call to [spinSystemReleaseInstance\(\)](#) to release.

14.14.1.26 spinVideo

```
typedef void* spinVideo
```

Handle for video recording functionality.

Created by calling [spinVideoOpenUncompressed\(\)](#), [spinVideoOpenMJPEG\(\)](#), and [spinVideoOpenH264\(\)](#), which require a call to [spinVideoClose\(\)](#) to destroy.

14.14.2 Enumeration Type Documentation

14.14.2.1 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 | |

14.14.2.2 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_NONE | No color processing. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_NEAREST_NEIGHBOR | Fastest but lowest quality. Equivalent to FLYCAPTURE_NEAREST_NEIGHBOR_FAST in FlyCapture. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_NEAREST_NEIGHBOR_AVG | Nearest Neighbor with averaged green pixels. Higher quality but slower compared to nearest neighbor without averaging. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_BILINEAR | Weighted average of surrounding 4 pixels in a 2x2 neighborhood. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_EDGE_SENSING | Weights surrounding pixels based on localized edge orientation. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_HQ_LINEAR | Well-balanced speed and quality. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_IPP | Multi-threaded with similar results to edge sensing. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_DIRECTIONAL_FILTER | Best quality but much faster than rigorous. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_RIGOROUS | Slowest but produces good results. |
| SPINNAKER_COLOR_PROCESSING_↔ ALGORITHM_WEIGHTED_DIRECTIONAL_FILTER | Weighted pixel average from different directions. |

14.14.2.3 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 | |

Enumerator

| | |
|--|--|
| 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 | |
| SPINNAKER_ERR_CUSTOM_ID | Error codes less than -10000 are reserved for user-defined custom errors. |

14.14.2.4 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. |

Enumerator

| | |
|--|----------------------------|
| 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 | |

14.14.2.5 spinImageStatus

enum [spinImageStatus](#)

Status of images returned from [spinImageGetStatus\(\)](#) call.

Enumerator

| | |
|---|---|
| SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR | Image has an unknown error. |
| SPINNAKER_IMAGE_STATUS_NO_ERROR | Image is returned from GetNextImage() call without any errors. |
| SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED | Image failed CRC check. |
| SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW | Received more data than the size of the image. |
| SPINNAKER_IMAGE_STATUS_MISSING_PACKETS | 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_BUFFER_SIZE_INCONSISTENT | Image leader is incomplete. Could be caused by missing packet(s). See link above. |
| 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_INCONSISTENT | 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. |

14.14.2.6 spinnakerLogLevel

enum `spinnakerLogLevel`

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 | |

14.14.2.7 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 | |

14.14.2.8 spinTIFFCompressionMethod

enum `spinTIFFCompressionMethod`

Compression method to use for encoding TIFF images.

Enumerator

| | |
|-------------------------------------|--|
| SPINNAKER_TIFF_COMPRESS_METHOD_NONE | |
|-------------------------------------|--|

Enumerator

| | |
|--|--|
| 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 | |

14.14.2.9 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 | |
| SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID | |
| SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED | |
| SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED | |
| SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED | |

14.14.2.10 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 | |

14.14.3 Variable Documentation

14.14.3.1 False

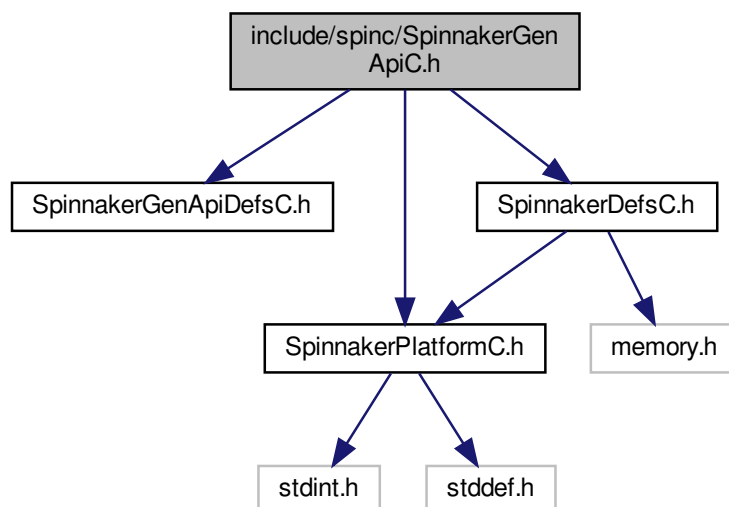
```
const bool8_t False = 0 [static]
```

14.14.3.2 True

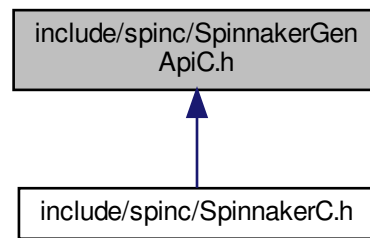
```
const bool8_t True = 1 [static]
```

14.15 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 spinNodeIsImplemented](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbResult)
Checks whether a node is implemented.
- [SPINNAKERC_API spinNodeIsReadable](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbResult)
Checks whether a node is readable.
- [SPINNAKERC_API spinNodeIsWritable](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbResult)
Checks whether a node is writable.
- [SPINNAKERC_API spinNodeIsAvailable](#) ([spinNodeHandle](#) hNode, [bool8_t](#) *pbResult)
Checks whether a node is available.
- [SPINNAKERC_API spinNodeIsEqual](#) ([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.

14.15.1 Function Documentation

14.15.1.1 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

14.15.1.2 `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

14.15.1.3 `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

14.15.1.4 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

14.15.1.5 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

14.15.1.6 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

14.15.1.7 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

14.15.1.8 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

14.15.1.9 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

14.15.1.10 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

14.15.1.11 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

14.15.1.12 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

14.15.1.13 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

14.15.1.14 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

14.15.1.15 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

14.15.1.16 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

14.15.1.17 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

14.15.1.18 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

14.15.1.19 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

14.15.1.20 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

14.15.1.21 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

14.15.1.22 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

14.15.1.23 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

14.15.1.24 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

14.15.1.25 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

14.15.1.26 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

14.15.1.27 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

14.15.1.28 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

14.15.1.29 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

14.15.1.30 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

14.15.1.31 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

14.15.1.32 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

14.15.1.33 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

14.15.1.34 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

14.15.1.35 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

14.15.1.36 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

14.15.1.37 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

14.15.1.38 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

14.15.1.39 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

14.15.1.40 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

14.15.1.41 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

14.15.1.42 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

14.15.1.43 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

14.15.1.44 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

14.15.1.45 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

14.15.1.46 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

14.15.1.47 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

14.15.1.48 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

14.15.1.49 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

14.15.1.50 spinNodeIsAvailable()

```
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

14.15.1.51 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

14.15.1.52 spinNodesImplemented()

```
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

14.15.1.53 spinNodesReadable()

```
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

14.15.1.54 spinNodesWritable()

```
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

14.15.1.55 `spinNodeMapGetNode()`

```
SPINNAKERC_API spinNodeMapGetNode (
    spinNodeMapHandle hNodeMap,
    const char * pName,
    spinNodeHandle * phNode )
```

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>phNode</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

14.15.1.56 `spinNodeMapGetNodeByIndex()`

```
SPINNAKERC_API spinNodeMapGetNodeByIndex (
    spinNodeMapHandle hNodeMap,
    size_t index,
    spinNodeHandle * phNode )
```

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>phNode</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

14.15.1.57 `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

14.15.1.58 `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

14.15.1.59 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

14.15.1.60 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

14.15.1.61 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

14.15.1.62 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

14.15.1.63 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

14.15.1.64 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

14.15.1.65 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

14.15.1.66 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

14.15.1.67 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

14.15.1.68 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

14.15.1.69 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

14.15.1.70 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

14.15.1.71 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

14.15.1.72 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

14.15.1.73 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

14.15.1.74 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

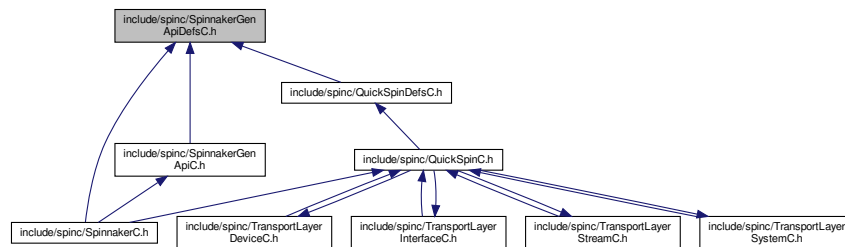
| | |
|----------------|---|
| <i>hNode</i> | The string node having its value changed |
| <i>bVerify</i> | The boolean of whether to verify the node |
| <i>pBuf</i> | The c-string of the value to set |

Returns

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

14.16 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` {
`intflValue` ,
`intflBase` ,
`intflInteger` ,
`intflBoolean` ,
`intflCommand` ,
`intflFloat` ,
`intflString` ,
`intflRegister` ,
`intflCategory` ,
`intflEnumeration` ,
`intflEnumEntry` ,
`intflPort` }
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

14.16.1 Typedef Documentation

14.16.1.1 spinNodeCallbackFunction

```
typedef void(* spinNodeCallbackFunction) (spinNodeHandle hNode)
```

Function signatures are used to create and trigger callbacks and events.

14.16.1.2 spinNodeCallbackHandle

```
typedef void* spinNodeCallbackHandle
```

Handle for callback functionality.

Created by calling [spinNodeRegisterCallback\(\)](#), which requires a call to [spinNodeUnregisterCallback\(\)](#) destroy.

14.16.1.3 spinNodeHandle

```
typedef void* spinNodeHandle
```

Handle for node functionality.

Created by calling [spinNodeMapGetNode\(\)](#). No need to release, clear, or destroy.

14.16.1.4 spinNodeMapHandle

```
typedef void* spinNodeMapHandle
```

Handle for nodemap functionality.

Created by calling [spinCameraGetNodemap\(\)](#), [spinCameraGetTLDeviceNodeMap\(\)](#), [spinCameraGetTLStreamNodeMap\(\)](#) or [spinInterfaceGetTLNodeMap\(\)](#). No need to release, clear, or destroy.

14.16.2 Enumeration Type Documentation

14.16.2.1 spinAccessMode

```
enum spinAccessMode
```

Enumerator

| | |
|-----------------------|--|
| NI | |
| NA | |
| WO | |
| RO | |
| RW | |
| _UndefinedAccesMode | |
| _CycleDetectAccesMode | |

14.16.2.2 spinCachingMode

```
enum spinCachingMode
```

Enumerator

| | |
|-----------------------|--|
| NoCache | |
| WriteThrough | |
| WriteAround | |
| _UndefinedCachingMode | |

14.16.2.3 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 |

14.16.2.4 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. |

14.16.2.5 spinIncMode

enum `spinIncMode`

typedef for increment mode

Enumerator

| | |
|----------------|--|
| noIncrement | |
| fixedIncrement | |
| listIncrement | |

14.16.2.6 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 |

14.16.2.7 spinInterfaceType

enum [spinInterfaceType](#)

typedef for interface type

Enumerator

| | |
|------------------|---|
| intflValue | |
| intflBase | IValue interface |
| intflInteger | IBase interface |
| intflBoolean | IInteger interface |
| intflCommand | IBoolean interface |
| intflFloat | ICommand interface |
| intflString | IFloat interface |
| intflRegister | IString interface |
| intflCategory | IRegister interface |
| intflEnumeration | ICategory interface |
| intflEnumEntry | IEnumeration interface |
| intflPort | IEnumEntry interface IPort interface |

14.16.2.8 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 |

14.16.2.9 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. |

14.16.2.10 spinNodeType

enum `spinNodeType`

Enumerator

| | |
|-------------|--|
| ValueNode | |
| BaseNode | |
| IntegerNode | |
| BooleanNode | |
| FloatNode | |
| CommandNode | |
| StringNode | |

Enumerator

| | |
|-----------------|--|
| RegisterNode | |
| EnumerationNode | |
| EnumEntryNode | |
| CategoryNode | |
| PortNode | |
| UnknownNode | |

14.16.2.11 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 | |

14.16.2.12 spinSign

enum [spinSign](#)

Enumerator

| | |
|----------------|--|
| Signed | |
| Unsigned | |
| _UndefinedSign | |

14.16.2.13 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 |

14.16.2.14 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. |

14.16.2.15 spinVisibility

enum `spinVisibility`

Enumerator

| | |
|----------------------|--|
| Beginner | |
| Expert | |
| Guru | |
| Invisible | |
| _UndefinedVisibility | |

14.16.2.16 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

| | |
|---------------------------------------|---|
| <code>xvLoad</code> | |
| <code>xvCycles</code> | Creates a dummy node map |
| <code>xvSFNC</code> | checks for write and dependency cycles (implies <code>xvLoad</code>) |
| <code>xvDefault</code> | checks for conformance with the standard feature naming convention (SFNC) |
| <code>xvAll</code> | checks performed if nothing else is said |
| <code>_UndefinedEXMLValidation</code> | all possible checks Object is not yet initialized |

14.16.2.17 spinYesNo

enum `spinYesNo`

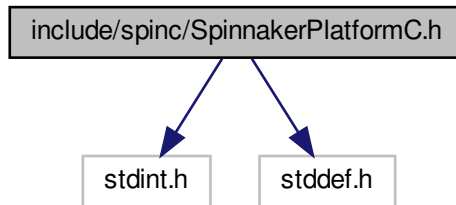
Defines the choices of a Yes/No alternative.

Enumerator

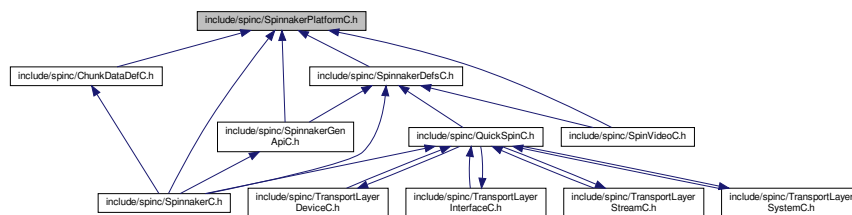
| | |
|------------------------------|--------------------------------|
| <code>Yes</code> | <code>yes</code> |
| <code>No</code> | <code>no</code> |
| <code>_UndefinedYesNo</code> | Object is not yet initialized. |

14.17 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`

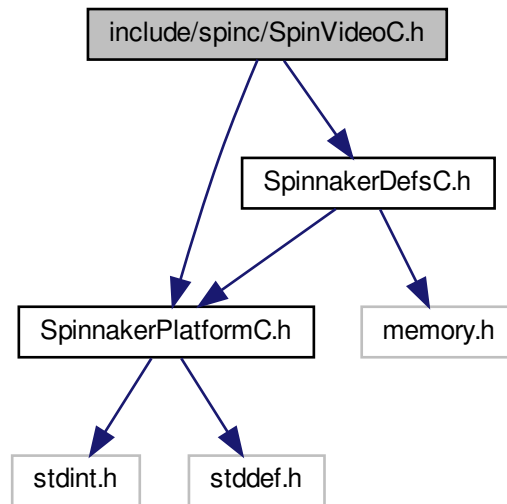
14.17.1 Macro Definition Documentation

14.17.1.1 SPINNAKERC_API

```
#define SPINNAKERC_API SPINC_IMPORT_EXPORT spinError SPINC_CALLTYPE
```

14.18 include/spinc/SpinVideoC.h File Reference

Include dependency graph for SpinVideoC.h:



Functions

- [SPINNAKERC_API spinVideoOpenUncompressed](#) ([spinVideo](#) *phSpinVideo, const char *pName, [spinAVIOption](#) 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)

14.18.1 Function Documentation

14.18.1.1 spinVideoAppend()

```

SPINNAKERC_API spinVideoAppend (
    spinVideo hSpinVideo,
    spinImage hImage )

```

14.18.1.2 spinVideoClose()

```
SPINNAKERC_API spinVideoClose (
    spinVideo hSpinVideo )
```

14.18.1.3 spinVideoOpenH264()

```
SPINNAKERC_API spinVideoOpenH264 (
    spinVideo * phSpinVideo,
    const char * pName,
    spinH264Option option )
```

14.18.1.4 spinVideoOpenMJPEG()

```
SPINNAKERC_API spinVideoOpenMJPEG (
    spinVideo * phSpinVideo,
    const char * pName,
    spinMJPEGOption option )
```

14.18.1.5 spinVideoOpenUncompressed()

```
SPINNAKERC_API spinVideoOpenUncompressed (
    spinVideo * phSpinVideo,
    const char * pName,
    spinAVIOption option )
```

14.18.1.6 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

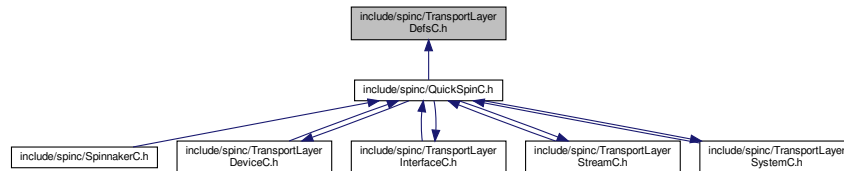
| | |
|-------------------|--|
| <i>hSpinVideo</i> | The spin video recorder to append the image to |
| <i>size</i> | The maximum video file size in MB. |

Returns

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

14.19 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_TeledyneGigeVision](#) ,
[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 `spinTLDeviceEndiannessMechanismEnums` {
`DeviceEndiannessMechanism_Legacy` ,
`DeviceEndiannessMechanism_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 `spinTLFLIRFilterDriverStatusEnums` {
`FLIRFilterDriverStatus_NotSupported` ,
`FLIRFilterDriverStatus_Disabled` ,
`FLIRFilterDriverStatus_Enabled` ,
`NUMFLIRFILTERDRIVERSTATUS` }
- enum `spinTLTeledyneGigeVisionFilterDriverStatusEnums` {
`TeledyneGigeVisionFilterDriverStatus_NotSupported` ,
`TeledyneGigeVisionFilterDriverStatus_Disabled` ,
`TeledyneGigeVisionFilterDriverStatus_Enabled` ,
`NUMTELEDYNEGIGEVISIONFILTERDRIVERSTATUS` }
- enum `spinTLTLTypeEnums` {
`TLType_GigEVision` ,

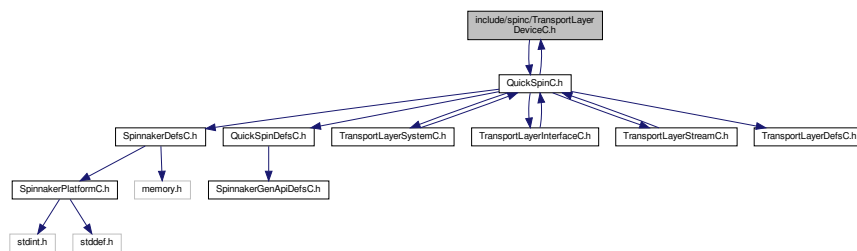

```

TLType_CameraLink ,
TLType_CameraLinkHS ,
TLType_CoaXPress ,
TLType_USB3Vision ,
TLType_Mixed ,
TLType_Custom ,
NUMTLTYPE }

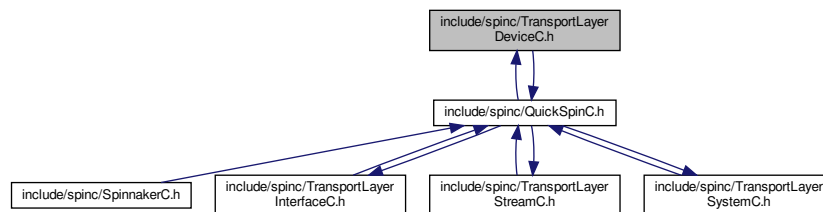
```

14.20 include/spinc/TransportLayerDeviceC.h File Reference

Include dependency graph for TransportLayerDeviceC.h:



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

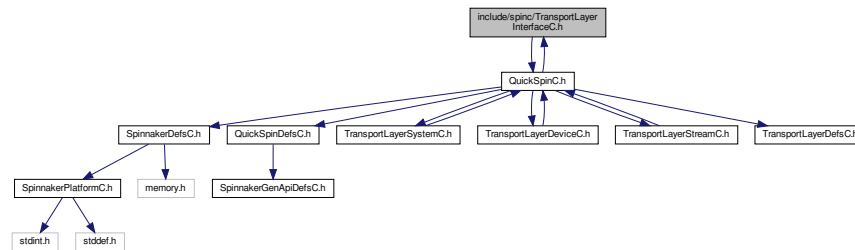


Data Structures

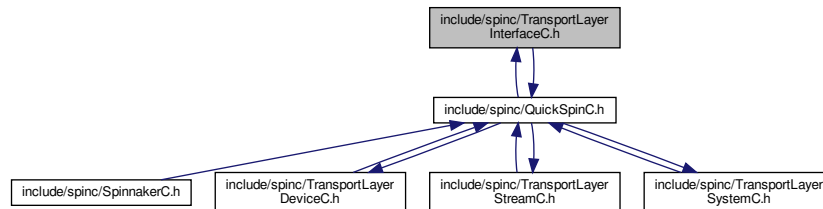
- struct [quickSpinTLDevice](#)

14.21 include/spinc/TransportLayerInterfaceC.h File Reference

Include dependency graph for TransportLayerInterfaceC.h:



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

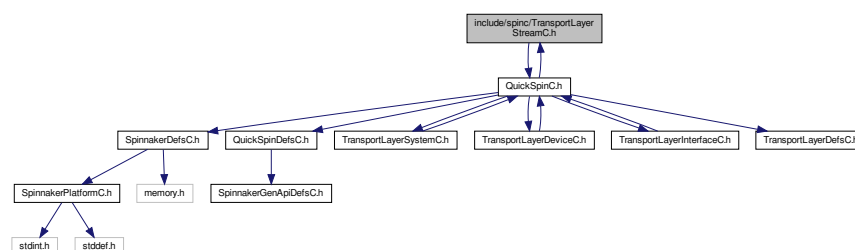


Data Structures

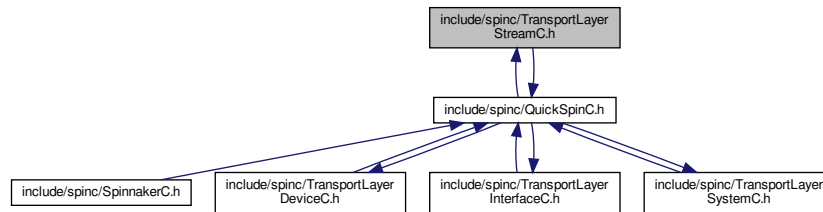
- struct [quickSpinTLInterface](#)

14.22 include/spinc/TransportLayerStreamC.h File Reference

Include dependency graph for TransportLayerStreamC.h:



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

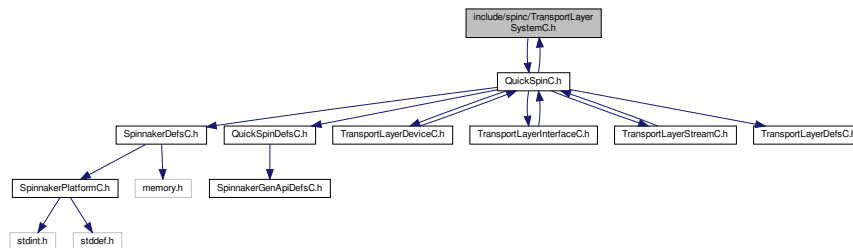


Data Structures

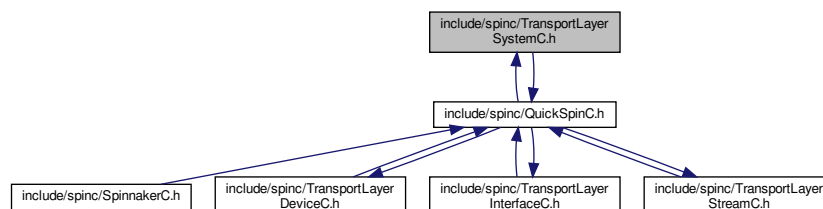
- struct [quickSpinTLStream](#)

14.23 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, [507](#)
 - [_UndefinedAccessMode](#)
SpinnakerGenApiDefsC.h, [507](#)
 - [_UndefinedCachingMode](#)
SpinnakerGenApiDefsC.h, [507](#)
 - [_UndefinedEDisplayNotation](#)
SpinnakerGenApiDefsC.h, [507](#)
 - [_UndefinedESlope](#)
SpinnakerGenApiDefsC.h, [512](#)
 - [_UndefinedEXMLValidation](#)
SpinnakerGenApiDefsC.h, [513](#)
 - [_UndefinedEndian](#)
SpinnakerGenApiDefsC.h, [508](#)
 - [_UndefinedNameSpace](#)
SpinnakerGenApiDefsC.h, [510](#)
 - [_UndefinedRepresentation](#)
SpinnakerGenApiDefsC.h, [511](#)
 - [_UndefinedSign](#)
SpinnakerGenApiDefsC.h, [511](#)
 - [_UndefinedStandardNameSpace](#)
SpinnakerGenApiDefsC.h, [512](#)
 - [_UndefinedVisibility](#)
SpinnakerGenApiDefsC.h, [512](#)
 - [_UndefinedYesNo](#)
SpinnakerGenApiDefsC.h, [513](#)
- [AasRoiEnable](#)
quickSpin, [174](#)
- [AasRoiHeight](#)
quickSpin, [174](#)
- [AasRoiOffsetX](#)
quickSpin, [174](#)
- [AasRoiOffsetY](#)
quickSpin, [174](#)
- [AasRoiWidth](#)
quickSpin, [174](#)
- [AcquisitionAbort](#)
quickSpin, [175](#)
- [AcquisitionArm](#)
quickSpin, [175](#)
- [AcquisitionBurstFrameCount](#)
quickSpin, [175](#)
- [AcquisitionFrameCount](#)
quickSpin, [175](#)
- [AcquisitionFrameRate](#)
quickSpin, [175](#)
- [AcquisitionFrameRateEnable](#)
quickSpin, [175](#)
- [AcquisitionLineRate](#)
quickSpin, [175](#)
- [AcquisitionMode](#)
quickSpin, [175](#)
- [AcquisitionMode_Continuous](#)
Camera Enumerations, [56](#)
- [AcquisitionMode_MultiFrame](#)
Camera Enumerations, [56](#)
- [AcquisitionMode_SingleFrame](#)
Camera Enumerations, [56](#)
- [AcquisitionResultingFrameRate](#)
quickSpin, [176](#)
- [AcquisitionStart](#)
quickSpin, [176](#)
- [AcquisitionStatus](#)
quickSpin, [176](#)
- [AcquisitionStatusSelector](#)
quickSpin, [176](#)
- [AcquisitionStatusSelector_AcquisitionActive](#)
Camera Enumerations, [57](#)
- [AcquisitionStatusSelector_AcquisitionTransfer](#)
Camera Enumerations, [57](#)
- [AcquisitionStatusSelector_AcquisitionTriggerWait](#)
Camera Enumerations, [57](#)
- [AcquisitionStatusSelector_ExposureActive](#)
Camera Enumerations, [57](#)
- [AcquisitionStatusSelector_FrameActive](#)
Camera Enumerations, [57](#)
- [AcquisitionStatusSelector_FrameTriggerWait](#)
Camera Enumerations, [57](#)
- [AcquisitionStop](#)
quickSpin, [176](#)
- [ActionCommand](#)
quickSpinTLInterface, [266](#)
- [actionCommandResult](#), [161](#)
DeviceAddress, [161](#)
Status, [161](#)
- [ActionDeviceKey](#)
quickSpin, [176](#)
- [ActionGroupKey](#)
quickSpin, [176](#)
- [ActionGroupMask](#)
quickSpin, [176](#)
- [ActionQueueSize](#)
quickSpin, [177](#)
- [ActionSelector](#)
quickSpin, [177](#)
- [ActionUnconditionalMode](#)
quickSpin, [177](#)
- [ActionUnconditionalMode_Off](#)

- Camera Enumerations, [57](#)
- ActionUnconditionalMode_On
 - Camera Enumerations, [57](#)
- AdaptiveCompressionEnable
 - quickSpin, [177](#)
- AdcBitDepth
 - quickSpin, [177](#)
- AdcBitDepth_Bit10
 - Camera Enumerations, [57](#)
- AdcBitDepth_Bit12
 - Camera Enumerations, [57](#)
- AdcBitDepth_Bit14
 - Camera Enumerations, [57](#)
- AdcBitDepth_Bit8
 - Camera Enumerations, [57](#)
- aPAUSEMACCtrlFramesReceived
 - quickSpin, [177](#)
- aPAUSEMACCtrlFramesTransmitted
 - quickSpin, [177](#)
- AutoAlgorithmSelector
 - quickSpin, [177](#)
- AutoAlgorithmSelector_Ae
 - Camera Enumerations, [58](#)
- AutoAlgorithmSelector_Awb
 - Camera Enumerations, [58](#)
- AutoExposureControlLoopDamping
 - quickSpin, [178](#)
- AutoExposureControlPriority
 - quickSpin, [178](#)
- AutoExposureControlPriority_ExposureTime
 - Camera Enumerations, [58](#)
- AutoExposureControlPriority_Gain
 - Camera Enumerations, [58](#)
- AutoExposureEVCompensation
 - quickSpin, [178](#)
- AutoExposureExposureTimeLowerLimit
 - quickSpin, [178](#)
- AutoExposureExposureTimeUpperLimit
 - quickSpin, [178](#)
- AutoExposureGainLowerLimit
 - quickSpin, [178](#)
- AutoExposureGainUpperLimit
 - quickSpin, [178](#)
- AutoExposureGreyValueLowerLimit
 - quickSpin, [178](#)
- AutoExposureGreyValueUpperLimit
 - quickSpin, [179](#)
- AutoExposureLightingMode
 - quickSpin, [179](#)
- AutoExposureLightingMode_AutoDetect
 - Camera Enumerations, [58](#)
- AutoExposureLightingMode_Backlight
 - Camera Enumerations, [58](#)
- AutoExposureLightingMode_Frontlight
 - Camera Enumerations, [58](#)
- AutoExposureLightingMode_Normal
 - Camera Enumerations, [58](#)
- AutoExposureMeteringMode
 - quickSpin, [179](#)
- AutoExposureMeteringMode_Average
 - Camera Enumerations, [59](#)
- AutoExposureMeteringMode_CenterWeighted
 - Camera Enumerations, [59](#)
- AutoExposureMeteringMode_HistogramPeak
 - Camera Enumerations, [59](#)
- AutoExposureMeteringMode_Partial
 - Camera Enumerations, [59](#)
- AutoExposureMeteringMode_Spot
 - Camera Enumerations, [59](#)
- AutoExposureTargetGreyValue
 - quickSpin, [179](#)
- AutoExposureTargetGreyValueAuto
 - quickSpin, [179](#)
- AutoExposureTargetGreyValueAuto_Continuous
 - Camera Enumerations, [59](#)
- AutoExposureTargetGreyValueAuto_Off
 - Camera Enumerations, [59](#)
- Automatic
 - SpinnakerGenApiDefsC.h, [512](#)
- BalanceRatio
 - quickSpin, [179](#)
- BalanceRatioSelector
 - quickSpin, [179](#)
- BalanceRatioSelector_Blue
 - Camera Enumerations, [60](#)
- BalanceRatioSelector_Red
 - Camera Enumerations, [60](#)
- BalanceWhiteAuto
 - quickSpin, [179](#)
- BalanceWhiteAuto_Continuous
 - Camera Enumerations, [60](#)
- BalanceWhiteAuto_Off
 - Camera Enumerations, [60](#)
- BalanceWhiteAuto_Once
 - Camera Enumerations, [60](#)
- BalanceWhiteAutoDamping
 - quickSpin, [180](#)
- BalanceWhiteAutoLowerLimit
 - quickSpin, [180](#)
- BalanceWhiteAutoProfile
 - quickSpin, [180](#)
- BalanceWhiteAutoProfile_Indoor
 - Camera Enumerations, [60](#)
- BalanceWhiteAutoProfile_Outdoor
 - Camera Enumerations, [60](#)
- BalanceWhiteAutoUpperLimit
 - quickSpin, [180](#)
- BaseNode
 - SpinnakerGenApiDefsC.h, [510](#)
- Beginner
 - SpinnakerGenApiDefsC.h, [512](#)
- BigEndian
 - SpinnakerGenApiDefsC.h, [508](#)
- binaryFile
 - spinPGMOption, [299](#)
 - spinPPMOption, [300](#)

- BinningHorizontal
 - quickSpin, [180](#)
- BinningHorizontalMode
 - quickSpin, [180](#)
- BinningHorizontalMode_Average
 - Camera Enumerations, [61](#)
- BinningHorizontalMode_Sum
 - Camera Enumerations, [61](#)
- BinningSelector
 - quickSpin, [180](#)
- BinningSelector_All
 - Camera Enumerations, [61](#)
- BinningSelector_ISP
 - Camera Enumerations, [61](#)
- BinningSelector_Sensor
 - Camera Enumerations, [61](#)
- BinningVertical
 - quickSpin, [180](#)
- BinningVerticalMode
 - quickSpin, [181](#)
- BinningVerticalMode_Average
 - Camera Enumerations, [61](#)
- BinningVerticalMode_Sum
 - Camera Enumerations, [61](#)
- bitrate
 - spinH264Option, [292](#)
- BlackLevel
 - quickSpin, [181](#)
- BlackLevelAuto
 - quickSpin, [181](#)
- BlackLevelAuto_Continuous
 - Camera Enumerations, [62](#)
- BlackLevelAuto_Off
 - Camera Enumerations, [62](#)
- BlackLevelAuto_Once
 - Camera Enumerations, [62](#)
- BlackLevelAutoBalance
 - quickSpin, [181](#)
- BlackLevelAutoBalance_Continuous
 - Camera Enumerations, [62](#)
- BlackLevelAutoBalance_Off
 - Camera Enumerations, [62](#)
- BlackLevelAutoBalance_Once
 - Camera Enumerations, [62](#)
- BlackLevelClampingEnable
 - quickSpin, [181](#)
- BlackLevelRaw
 - quickSpin, [181](#)
- BlackLevelSelector
 - quickSpin, [181](#)
- BlackLevelSelector_All
 - Camera Enumerations, [62](#)
- BlackLevelSelector_Analog
 - Camera Enumerations, [62](#)
- BlackLevelSelector_Digital
 - Camera Enumerations, [62](#)
- bool8_t
 - SpinnakerDefsC.h, [452](#)
- Boolean
 - SpinnakerGenApiDefsC.h, [511](#)
- BooleanNode
 - SpinnakerGenApiDefsC.h, [510](#)
- build
 - spinLibraryVersion, [296](#)
- Camera Access, [143](#)
- Camera Enumerations, [24](#)
 - AcquisitionMode_Continuous, [56](#)
 - AcquisitionMode_MultiFrame, [56](#)
 - AcquisitionMode_SingleFrame, [56](#)
 - AcquisitionStatusSelector_AcquisitionActive, [57](#)
 - AcquisitionStatusSelector_AcquisitionTransfer, [57](#)
 - AcquisitionStatusSelector_AcquisitionTriggerWait, [57](#)
 - AcquisitionStatusSelector_ExposureActive, [57](#)
 - AcquisitionStatusSelector_FrameActive, [57](#)
 - AcquisitionStatusSelector_FrameTriggerWait, [57](#)
 - ActionUnconditionalMode_Off, [57](#)
 - ActionUnconditionalMode_On, [57](#)
 - AdcBitDepth_Bit10, [57](#)
 - AdcBitDepth_Bit12, [57](#)
 - AdcBitDepth_Bit14, [57](#)
 - AdcBitDepth_Bit8, [57](#)
 - AutoAlgorithmSelector_Ae, [58](#)
 - AutoAlgorithmSelector_Awb, [58](#)
 - AutoExposureControlPriority_ExposureTime, [58](#)
 - AutoExposureControlPriority_Gain, [58](#)
 - AutoExposureLightingMode_AutoDetect, [58](#)
 - AutoExposureLightingMode_Backlight, [58](#)
 - AutoExposureLightingMode_Frontlight, [58](#)
 - AutoExposureLightingMode_Normal, [58](#)
 - AutoExposureMeteringMode_Average, [59](#)
 - AutoExposureMeteringMode_CenterWeighted, [59](#)
 - AutoExposureMeteringMode_HistogramPeak, [59](#)
 - AutoExposureMeteringMode_Partial, [59](#)
 - AutoExposureMeteringMode_Spot, [59](#)
 - AutoExposureTargetGreyValueAuto_Continuous, [59](#)
 - AutoExposureTargetGreyValueAuto_Off, [59](#)
 - BalanceRatioSelector_Blue, [60](#)
 - BalanceRatioSelector_Red, [60](#)
 - BalanceWhiteAuto_Continuous, [60](#)
 - BalanceWhiteAuto_Off, [60](#)
 - BalanceWhiteAuto_Once, [60](#)
 - BalanceWhiteAutoProfile_Indoor, [60](#)
 - BalanceWhiteAutoProfile_Outdoor, [60](#)
 - BinningHorizontalMode_Average, [61](#)
 - BinningHorizontalMode_Sum, [61](#)
 - BinningSelector_All, [61](#)
 - BinningSelector_ISP, [61](#)
 - BinningSelector_Sensor, [61](#)
 - BinningVerticalMode_Average, [61](#)
 - BinningVerticalMode_Sum, [61](#)
 - BlackLevelAuto_Continuous, [62](#)
 - BlackLevelAuto_Off, [62](#)
 - BlackLevelAuto_Once, [62](#)
 - BlackLevelAutoBalance_Continuous, [62](#)

- BlackLevelAutoBalance_Off, 62
- BlackLevelAutoBalance_Once, 62
- BlackLevelSelector_All, 62
- BlackLevelSelector_Analog, 62
- BlackLevelSelector_Digital, 62
- ChunkBlackLevelSelector_All, 63
- ChunkCounterSelector_Counter0, 63
- ChunkCounterSelector_Counter1, 63
- ChunkCounterSelector_Counter2, 63
- ChunkEncoderSelector_Encoder0, 63
- ChunkEncoderSelector_Encoder1, 63
- ChunkEncoderSelector_Encoder2, 63
- ChunkEncoderStatus_EncoderDown, 63
- ChunkEncoderStatus_EncoderIdle, 63
- ChunkEncoderStatus_EncoderStatic, 63
- ChunkEncoderStatus_EncoderUp, 63
- ChunkExposureTimeSelector_Blue, 64
- ChunkExposureTimeSelector_Common, 64
- ChunkExposureTimeSelector_Cyan, 64
- ChunkExposureTimeSelector_Green, 64
- ChunkExposureTimeSelector_Infrared, 64
- ChunkExposureTimeSelector_Magenta, 64
- ChunkExposureTimeSelector_Red, 64
- ChunkExposureTimeSelector_Stage1, 64
- ChunkExposureTimeSelector_Stage2, 64
- ChunkExposureTimeSelector_Ultraviolet, 64
- ChunkExposureTimeSelector_Yellow, 64
- ChunkGainSelector_All, 64
- ChunkGainSelector_Blue, 64
- ChunkGainSelector_Green, 64
- ChunkGainSelector_Red, 64
- ChunkImageComponent_Color, 65
- ChunkImageComponent_Confidence, 65
- ChunkImageComponent_Disparity, 65
- ChunkImageComponent_Infrared, 65
- ChunkImageComponent_Intensity, 65
- ChunkImageComponent_Range, 65
- ChunkImageComponent_Scatter, 65
- ChunkImageComponent_Ultraviolet, 65
- ChunkPixelFormat_BayerBG8, 65
- ChunkPixelFormat_BayerGB8, 65
- ChunkPixelFormat_BayerGR8, 65
- ChunkPixelFormat_BayerRG8, 65
- ChunkPixelFormat_Mono12Packed, 65
- ChunkPixelFormat_Mono16, 65
- ChunkPixelFormat_Mono8, 65
- ChunkPixelFormat_RGB8Packed, 65
- ChunkPixelFormat_YCbCr601_422_8_CbYCrY, 65
- ChunkPixelFormat_YUV422Packed, 65
- ChunkRegionID_Region0, 65
- ChunkRegionID_Region1, 65
- ChunkRegionID_Region2, 65
- ChunkScan3dCoordinateReferenceSelector_RotationX, 66
- ChunkScan3dCoordinateReferenceSelector_RotationY, 66
- ChunkScan3dCoordinateReferenceSelector_RotationZ, 66
- ChunkScan3dCoordinateReferenceSelector_TranslationX, 66
- ChunkScan3dCoordinateReferenceSelector_TranslationY, 66
- ChunkScan3dCoordinateReferenceSelector_TranslationZ, 66
- ChunkScan3dCoordinateSelector_CoordinateA, 66
- ChunkScan3dCoordinateSelector_CoordinateB, 66
- ChunkScan3dCoordinateSelector_CoordinateC, 66
- ChunkScan3dCoordinateSystem_Cartesian, 66
- ChunkScan3dCoordinateSystem_Cylindrical, 66
- ChunkScan3dCoordinateSystem_Spherical, 66
- ChunkScan3dCoordinateSystemReference_Anchor, 67
- ChunkScan3dCoordinateSystemReference_Transformed, 67
- ChunkScan3dCoordinateTransformSelector_RotationX, 67
- ChunkScan3dCoordinateTransformSelector_RotationY, 67
- ChunkScan3dCoordinateTransformSelector_RotationZ, 67
- ChunkScan3dCoordinateTransformSelector_TranslationX, 67
- ChunkScan3dCoordinateTransformSelector_TranslationY, 67
- ChunkScan3dCoordinateTransformSelector_TranslationZ, 67
- ChunkScan3dDistanceUnit_Inch, 67
- ChunkScan3dDistanceUnit_Millimeter, 67
- ChunkScan3dOutputMode_CalibratedABC_Grid, 68
- ChunkScan3dOutputMode_CalibratedABC_PointCloud, 68
- ChunkScan3dOutputMode_CalibratedAC, 68
- ChunkScan3dOutputMode_CalibratedAC_Linescan, 68
- ChunkScan3dOutputMode_CalibratedC, 68
- ChunkScan3dOutputMode_CalibratedC_Linescan, 68
- ChunkScan3dOutputMode_DisparityC, 68
- ChunkScan3dOutputMode_DisparityC_Linescan, 68
- ChunkScan3dOutputMode_RectifiedC, 68
- ChunkScan3dOutputMode_RectifiedC_Linescan, 68
- ChunkScan3dOutputMode_UncalibratedC, 68
- ChunkSelector_BlackLevel, 69
- ChunkSelector_CRC, 69
- ChunkSelector_ExposureEndLineStatusAll, 69
- ChunkSelector_ExposureTime, 69
- ChunkSelector_FrameID, 69
- ChunkSelector_Gain, 69

ChunkSelector_Height, 69
ChunkSelector_Image, 69
ChunkSelector_OffsetX, 69
ChunkSelector_OffsetY, 69
ChunkSelector_PixelFormat, 69
ChunkSelector_SequencerSetActive, 69
ChunkSelector_SerialData, 69
ChunkSelector_Timestamp, 69
ChunkSelector_Width, 69
ChunkSourceID_Source0, 69
ChunkSourceID_Source1, 69
ChunkSourceID_Source2, 69
ChunkTimerSelector_Timer0, 70
ChunkTimerSelector_Timer1, 70
ChunkTimerSelector_Timer2, 70
ChunkTransferStreamID_Stream0, 70
ChunkTransferStreamID_Stream1, 70
ChunkTransferStreamID_Stream2, 70
ChunkTransferStreamID_Stream3, 70
CIConfiguration_Base, 70
CIConfiguration_DualBase, 70
CIConfiguration_EightyBit, 70
CIConfiguration_Full, 70
CIConfiguration_Medium, 70
CITimeSlotsCount_One, 71
CITimeSlotsCount_Three, 71
CITimeSlotsCount_Two, 71
ColorTransformationSelector_RGBtoRGB, 71
ColorTransformationSelector_RGBtoYUV, 71
ColorTransformationValueSelector_Gain00, 71
ColorTransformationValueSelector_Gain01, 71
ColorTransformationValueSelector_Gain02, 71
ColorTransformationValueSelector_Gain10, 71
ColorTransformationValueSelector_Gain11, 71
ColorTransformationValueSelector_Gain12, 71
ColorTransformationValueSelector_Gain20, 71
ColorTransformationValueSelector_Gain21, 71
ColorTransformationValueSelector_Gain22, 71
ColorTransformationValueSelector_Offset0, 71
ColorTransformationValueSelector_Offset1, 71
ColorTransformationValueSelector_Offset2, 71
CompressionSaturationPriority_DropFrame, 72
CompressionSaturationPriority_ReduceFrameRate, 72
CounterEventActivation_AnyEdge, 72
CounterEventActivation_FallingEdge, 72
CounterEventActivation_LevelHigh, 72
CounterEventActivation_LevelLow, 72
CounterEventActivation_RisingEdge, 72
CounterEventSource_Counter0End, 73
CounterEventSource_Counter0Start, 73
CounterEventSource_Counter1End, 73
CounterEventSource_Counter1Start, 73
CounterEventSource_ExposureEnd, 73
CounterEventSource_ExposureStart, 73
CounterEventSource_FrameTriggerWait, 73
CounterEventSource_Line0, 72
CounterEventSource_Line1, 73
CounterEventSource_Line2, 73
CounterEventSource_Line3, 73
CounterEventSource_LogicBlock0, 73
CounterEventSource_LogicBlock1, 73
CounterEventSource_MHzTick, 72
CounterEventSource_Off, 72
CounterEventSource_UserOutput0, 73
CounterEventSource_UserOutput1, 73
CounterEventSource_UserOutput2, 73
CounterEventSource_UserOutput3, 73
CounterResetActivation_AnyEdge, 73
CounterResetActivation_FallingEdge, 73
CounterResetActivation_LevelHigh, 73
CounterResetActivation_LevelLow, 73
CounterResetActivation_RisingEdge, 73
CounterResetSource_Counter0End, 74
CounterResetSource_Counter0Start, 74
CounterResetSource_Counter1End, 74
CounterResetSource_Counter1Start, 74
CounterResetSource_ExposureEnd, 74
CounterResetSource_ExposureStart, 74
CounterResetSource_FrameTriggerWait, 74
CounterResetSource_Line0, 74
CounterResetSource_Line1, 74
CounterResetSource_Line2, 74
CounterResetSource_Line3, 74
CounterResetSource_LogicBlock0, 74
CounterResetSource_LogicBlock1, 74
CounterResetSource_Off, 73
CounterResetSource_UserOutput0, 74
CounterResetSource_UserOutput1, 74
CounterResetSource_UserOutput2, 74
CounterResetSource_UserOutput3, 74
CounterSelector_Counter0, 74
CounterSelector_Counter1, 74
CounterStatus_CounterActive, 74
CounterStatus_CounterCompleted, 74
CounterStatus_CounterIdle, 74
CounterStatus_CounterOverflow, 74
CounterStatus_CounterTriggerWait, 74
CounterTriggerActivation_AnyEdge, 75
CounterTriggerActivation_FallingEdge, 75
CounterTriggerActivation_LevelHigh, 75
CounterTriggerActivation_LevelLow, 75
CounterTriggerActivation_RisingEdge, 75
CounterTriggerSource_Counter0End, 75
CounterTriggerSource_Counter0Start, 75
CounterTriggerSource_Counter1End, 75
CounterTriggerSource_Counter1Start, 75
CounterTriggerSource_ExposureEnd, 75
CounterTriggerSource_ExposureStart, 75
CounterTriggerSource_FrameTriggerWait, 75
CounterTriggerSource_Line0, 75
CounterTriggerSource_Line1, 75
CounterTriggerSource_Line2, 75
CounterTriggerSource_Line3, 75
CounterTriggerSource_LogicBlock0, 75
CounterTriggerSource_LogicBlock1, 75

- DeviceClockSelector_Sensor, 81
- DeviceClockSelector_SensorDigitization, 81
- DeviceConnectionStatus_Active, 81
- DeviceConnectionStatus_Inactive, 81
- DeviceIndicatorMode_Active, 82
- DeviceIndicatorMode_ErrorStatus, 82
- DeviceIndicatorMode_Inactive, 82
- DeviceLinkHeartbeatMode_Off, 82
- DeviceLinkHeartbeatMode_On, 82
- DeviceLinkThroughputLimitMode_Off, 82
- DeviceLinkThroughputLimitMode_On, 82
- DevicePowerSupplySelector_External, 83
- DeviceRegistersEndianness_Big, 83
- DeviceRegistersEndianness_Little, 83
- DeviceScanType_Areascan, 83
- DeviceSerialPortBaudRate_Baud115200, 84
- DeviceSerialPortBaudRate_Baud19200, 84
- DeviceSerialPortBaudRate_Baud230400, 84
- DeviceSerialPortBaudRate_Baud38400, 84
- DeviceSerialPortBaudRate_Baud460800, 84
- DeviceSerialPortBaudRate_Baud57600, 84
- DeviceSerialPortBaudRate_Baud921600, 84
- DeviceSerialPortBaudRate_Baud9600, 84
- DeviceSerialPortSelector_CameraLink, 84
- DeviceStreamChannelEndianness_Big, 84
- DeviceStreamChannelEndianness_Little, 84
- DeviceStreamChannelType_Receiver, 85
- DeviceStreamChannelType_Transmitter, 85
- DeviceTapGeometry_Geometry_10X, 86
- DeviceTapGeometry_Geometry_10X_1Y, 86
- DeviceTapGeometry_Geometry_1X, 85
- DeviceTapGeometry_Geometry_1X10, 86
- DeviceTapGeometry_Geometry_1X10_1Y, 86
- DeviceTapGeometry_Geometry_1X2, 85
- DeviceTapGeometry_Geometry_1X2_1Y, 85
- DeviceTapGeometry_Geometry_1X2_1Y2, 85
- DeviceTapGeometry_Geometry_1X2_2YE, 85
- DeviceTapGeometry_Geometry_1X3, 85
- DeviceTapGeometry_Geometry_1X3_1Y, 85
- DeviceTapGeometry_Geometry_1X4, 85
- DeviceTapGeometry_Geometry_1X4_1Y, 85
- DeviceTapGeometry_Geometry_1X8, 86
- DeviceTapGeometry_Geometry_1X8_1Y, 86
- DeviceTapGeometry_Geometry_1X_1Y, 85
- DeviceTapGeometry_Geometry_1X_1Y2, 85
- DeviceTapGeometry_Geometry_1X_2YE, 85
- DeviceTapGeometry_Geometry_2X, 85
- DeviceTapGeometry_Geometry_2X2, 85
- DeviceTapGeometry_Geometry_2X2_1Y, 85
- DeviceTapGeometry_Geometry_2X2E, 85
- DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M, 85
- DeviceTapGeometry_Geometry_2X2E_2YE, 86
- DeviceTapGeometry_Geometry_2X2M, 86
- DeviceTapGeometry_Geometry_2X_1Y, 85
- DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y, 85
- DeviceTapGeometry_Geometry_2X_2YE, 85
- DeviceTapGeometry_Geometry_2XE, 85
- DeviceTapGeometry_Geometry_2XE_1Y2, 85
- DeviceTapGeometry_Geometry_2XE_2YE, 85
- DeviceTapGeometry_Geometry_2XM, 85
- DeviceTapGeometry_Geometry_2XM_1Y, 85
- DeviceTapGeometry_Geometry_2XM_1Y2, 85
- DeviceTapGeometry_Geometry_2XM_2YE, 85
- DeviceTapGeometry_Geometry_3X, 85
- DeviceTapGeometry_Geometry_3X_1Y, 85
- DeviceTapGeometry_Geometry_4X, 85
- DeviceTapGeometry_Geometry_4X2, 86
- DeviceTapGeometry_Geometry_4X2_1Y, 86
- DeviceTapGeometry_Geometry_4X2E, 86
- DeviceTapGeometry_Geometry_4X2E_1Y, 86
- DeviceTapGeometry_Geometry_4X_1Y, 85
- DeviceTapGeometry_Geometry_8X, 86
- DeviceTapGeometry_Geometry_8X_1Y, 86
- DeviceTemperatureSelector_Sensor, 86
- DeviceTLType_CameraLink, 86
- DeviceTLType_CameraLinkHS, 86
- DeviceTLType_CoaXPress, 86
- DeviceTLType_Custom, 86
- DeviceTLType_GigEVision, 86
- DeviceTLType_USB3Vision, 86
- DeviceType_Peripheral, 87
- DeviceType_Receiver, 87
- DeviceType_Transceiver, 87
- DeviceType_Transmitter, 87
- EncoderMode_FourPhase, 87
- EncoderMode_HighResolution, 87
- EncoderOutputMode_DirectionDown, 88
- EncoderOutputMode_DirectionUp, 88
- EncoderOutputMode_Motion, 88
- EncoderOutputMode_Off, 87
- EncoderOutputMode_PositionDown, 88
- EncoderOutputMode_PositionUp, 87
- EncoderResetActivation_AnyEdge, 88
- EncoderResetActivation_FallingEdge, 88
- EncoderResetActivation_LevelHigh, 88
- EncoderResetActivation_LevelLow, 88
- EncoderResetActivation_RisingEdge, 88
- EncoderResetSource_AcquisitionEnd, 88
- EncoderResetSource_AcquisitionStart, 88
- EncoderResetSource_AcquisitionTrigger, 88
- EncoderResetSource_Action0, 89
- EncoderResetSource_Action1, 89
- EncoderResetSource_Action2, 89
- EncoderResetSource_Counter0End, 89
- EncoderResetSource_Counter0Start, 89
- EncoderResetSource_Counter1End, 89
- EncoderResetSource_Counter1Start, 89
- EncoderResetSource_Counter2End, 89
- EncoderResetSource_Counter2Start, 89
- EncoderResetSource_ExposureEnd, 89
- EncoderResetSource_ExposureStart, 88
- EncoderResetSource_FrameEnd, 88
- EncoderResetSource_FrameStart, 88
- EncoderResetSource_FrameTrigger, 88

- EncoderResetSource_Line0, 89
- EncoderResetSource_Line1, 89
- EncoderResetSource_Line2, 89
- EncoderResetSource_LinkTrigger0, 89
- EncoderResetSource_LinkTrigger1, 89
- EncoderResetSource_LinkTrigger2, 89
- EncoderResetSource_Off, 88
- EncoderResetSource_SoftwareSignal0, 89
- EncoderResetSource_SoftwareSignal1, 89
- EncoderResetSource_SoftwareSignal2, 89
- EncoderResetSource_Timer0End, 89
- EncoderResetSource_Timer0Start, 89
- EncoderResetSource_Timer1End, 89
- EncoderResetSource_Timer1Start, 89
- EncoderResetSource_Timer2End, 89
- EncoderResetSource_Timer2Start, 89
- EncoderResetSource_UserOutput0, 89
- EncoderResetSource_UserOutput1, 89
- EncoderResetSource_UserOutput2, 89
- EncoderSelector_Encoder0, 89
- EncoderSelector_Encoder1, 89
- EncoderSelector_Encoder2, 89
- EncoderSourceA_Line0, 90
- EncoderSourceA_Line1, 90
- EncoderSourceA_Line2, 90
- EncoderSourceA_Off, 90
- EncoderSourceB_Line0, 90
- EncoderSourceB_Line1, 90
- EncoderSourceB_Line2, 90
- EncoderSourceB_Off, 90
- EncoderStatus_EncoderDown, 90
- EncoderStatus_EncoderIdle, 90
- EncoderStatus_EncoderStatic, 90
- EncoderStatus_EncoderUp, 90
- EventNotification_Off, 91
- EventNotification_On, 91
- EventSelector_Error, 91
- EventSelector_ExposureEnd, 91
- EventSelector_SerialPortReceive, 91
- ExposureActiveMode_AllPixels, 91
- ExposureActiveMode_AnyPixels, 91
- ExposureActiveMode_Line1, 91
- ExposureAuto_Continuous, 92
- ExposureAuto_Off, 92
- ExposureAuto_Once, 92
- ExposureMode_Timed, 92
- ExposureMode_TriggerWidth, 92
- ExposureTimeMode_Common, 92
- ExposureTimeMode_Individual, 92
- ExposureTimeSelector_Blue, 93
- ExposureTimeSelector_Common, 93
- ExposureTimeSelector_Cyan, 93
- ExposureTimeSelector_Green, 93
- ExposureTimeSelector_Infrared, 93
- ExposureTimeSelector_Magenta, 93
- ExposureTimeSelector_Red, 93
- ExposureTimeSelector_Stage1, 93
- ExposureTimeSelector_Stage2, 93
- ExposureTimeSelector_Ultraviolet, 93
- ExposureTimeSelector_Yellow, 93
- FileOpenMode_Read, 93
- FileOpenMode_ReadWrite, 93
- FileOpenMode_Write, 93
- FileOperationSelector_Close, 94
- FileOperationSelector_Delete, 94
- FileOperationSelector_Open, 94
- FileOperationSelector_Read, 94
- FileOperationSelector_Write, 94
- FileOperationStatus_Failure, 94
- FileOperationStatus_Overflow, 94
- FileOperationStatus_Success, 94
- FileSelector_SerialPort0, 94
- FileSelector_UserFile1, 94
- FileSelector_UserSet0, 94
- FileSelector_UserSet1, 94
- FileSelector_UserSetDefault, 94
- GainAuto_Continuous, 96
- GainAuto_Off, 96
- GainAuto_Once, 96
- GainAutoBalance_Continuous, 96
- GainAutoBalance_Off, 96
- GainAutoBalance_Once, 96
- GainSelector_All, 96
- GevCCP_ControlAccess, 97
- GevCCP_ExclusiveAccess, 97
- GevCCP_OpenAccess, 97
- GevCurrentPhysicalLinkConfiguration_DynamicLAG, 97
- GevCurrentPhysicalLinkConfiguration_MultiLink, 97
- GevCurrentPhysicalLinkConfiguration_SingleLink, 97
- GevCurrentPhysicalLinkConfiguration_StaticLAG, 97
- GevGVCPExtendedStatusCodesSelector_Version1_1, 97
- GevGVCPExtendedStatusCodesSelector_Version2_0, 97
- GevGVSPExtendedIDMode_Off, 98
- GevGVSPExtendedIDMode_On, 98
- GevIEEE1588ClockAccuracy_Unknown, 98
- GevIEEE1588Mode_Auto, 98
- GevIEEE1588Mode_SlaveOnly, 98
- GevIEEE1588Status_Disabled, 98
- GevIEEE1588Status_Faulty, 98
- GevIEEE1588Status_Initializing, 98
- GevIEEE1588Status_Listening, 99
- GevIEEE1588Status_Master, 99
- GevIEEE1588Status_Passive, 99
- GevIEEE1588Status_PreMaster, 99
- GevIEEE1588Status_Slave, 99
- GevIEEE1588Status_Uncalibrated, 99
- GevIPConfigurationStatus_DHCP, 99
- GevIPConfigurationStatus_ForceIP, 99
- GevIPConfigurationStatus_LLA, 99
- GevIPConfigurationStatus_None, 99

- GevIPConfigurationStatus_PersistentIP, 99
- GevPhysicalLinkConfiguration_DynamicLAG, 99
- GevPhysicalLinkConfiguration_MultiLink, 99
- GevPhysicalLinkConfiguration_SingleLink, 99
- GevPhysicalLinkConfiguration_StaticLAG, 99
- GevSupportedOptionSelector_Action, 100
- GevSupportedOptionSelector_CCPApplicationSocket, 100
- GevSupportedOptionSelector_CommandsConcatenation, 100
- GevSupportedOptionSelector_DiscoveryAckDelay, 100
- GevSupportedOptionSelector_DiscoveryAckDelayWritable, 100
- GevSupportedOptionSelector_Event, 100
- GevSupportedOptionSelector_EventData, 100
- GevSupportedOptionSelector_ExtendedStatusCodes, 100
- GevSupportedOptionSelector_HeartbeatDisable, 100
- GevSupportedOptionSelector_IPConfigurationDHCP, 100
- GevSupportedOptionSelector_IPConfigurationLLA, 100
- GevSupportedOptionSelector_IPConfigurationPersistentIP, 100
- GevSupportedOptionSelector_LinkSpeed, 100
- GevSupportedOptionSelector_ManifestTable, 100
- GevSupportedOptionSelector_MessageChannelSourceSocket, 100
- GevSupportedOptionSelector_PacketResend, 100
- GevSupportedOptionSelector_PendingAck, 100
- GevSupportedOptionSelector_SerialNumber, 100
- GevSupportedOptionSelector_StreamChannelSourceSocket, 100
- GevSupportedOptionSelector_TestData, 100
- GevSupportedOptionSelector_UserDefinedName, 100
- GevSupportedOptionSelector_WriteMem, 100
- ImageComponentSelector_Color, 100
- ImageComponentSelector_Confidence, 101
- ImageComponentSelector_Disparity, 101
- ImageComponentSelector_Infrared, 100
- ImageComponentSelector_Intensity, 100
- ImageComponentSelector_Range, 100
- ImageComponentSelector_Scatter, 101
- ImageComponentSelector_Ultraviolet, 100
- ImageCompressionJPEGFormatOption_BaselineOptimized, 101
- ImageCompressionJPEGFormatOption_BaselineStandard, 101
- ImageCompressionJPEGFormatOption_Lossless, 101
- ImageCompressionJPEGFormatOption_Progressive, 101
- ImageCompressionMode_Lossless, 102
- ImageCompressionMode_Off, 102
- ImageCompressionRateOption_FixBitrate, 102
- ImageCompressionRateOption_FixQuality, 102
- LineFormat_LVDS, 102
- LineFormat_NoConnect, 102
- LineFormat_OpenDrain, 102
- LineFormat_OptoCoupled, 102
- LineFormat_RS422, 102
- LineFormat_TriState, 102
- LineFormat_TTL, 102
- LineInputFilterSelector_Debounce, 103
- LineInputFilterSelector_Deglitch, 103
- LineMode_Input, 103
- LineMode_Output, 103
- LineSelector_Line0, 103
- LineSelector_Line1, 103
- LineSelector_Line2, 103
- LineSelector_Line3, 103
- LineSource_AllPixel, 104
- LineSource_AnyPixel, 104
- LineSource_Counter0Active, 104
- LineSource_Counter1Active, 104
- LineSource_ExposureActive, 104
- LineSource_FrameTriggerWait, 104
- LineSource_Line0, 104
- LineSource_Line1, 104
- LineSource_Line2, 104
- LineSource_Line3, 104
- LineSource_LogicBlock0, 104
- LineSource_LogicBlock1, 104
- LineSource_Off, 104
- LineSource_PPSSignal, 104
- LineSource_SerialPort0, 104
- LineSource_UserOutput0, 104
- LineSource_UserOutput1, 104
- LineSource_UserOutput2, 104
- LineSource_UserOutput3, 104
- LogicBlockLUTInputActivation_AnyEdge, 104
- LogicBlockLUTInputActivation_FallingEdge, 104
- LogicBlockLUTInputActivation_LevelHigh, 104
- LogicBlockLUTInputActivation_LevelLow, 104
- LogicBlockLUTInputActivation_RisingEdge, 104
- LogicBlockLUTInputSelector_Input0, 105
- LogicBlockLUTInputSelector_Input1, 105
- LogicBlockLUTInputSelector_Input2, 105
- LogicBlockLUTInputSelector_Input3, 105
- LogicBlockLUTInputSource_AcquisitionActive, 105
- LogicBlockLUTInputSource_Counter0End, 105
- LogicBlockLUTInputSource_Counter0Start, 105
- LogicBlockLUTInputSource_Counter1End, 105
- LogicBlockLUTInputSource_Counter1Start, 105
- LogicBlockLUTInputSource_ExposureEnd, 105
- LogicBlockLUTInputSource_ExposureStart, 105
- LogicBlockLUTInputSource_FrameTriggerWait, 105
- LogicBlockLUTInputSource_Line0, 105
- LogicBlockLUTInputSource_Line1, 105
- LogicBlockLUTInputSource_Line2, 105
- LogicBlockLUTInputSource_Line3, 105
- LogicBlockLUTInputSource_LogicBlock0, 105

- LogicBlockLUTInputSource_LogicBlock1, 105
- LogicBlockLUTInputSource_UserOutput0, 105
- LogicBlockLUTInputSource_UserOutput1, 105
- LogicBlockLUTInputSource_UserOutput2, 105
- LogicBlockLUTInputSource_UserOutput3, 105
- LogicBlockLUTInputSource_Zero, 105
- LogicBlockLUTSelector_Enable, 106
- LogicBlockLUTSelector_Value, 106
- LogicBlockSelector_LogicBlock0, 106
- LogicBlockSelector_LogicBlock1, 106
- LUTSelector_LUT1, 106
- NUM_ACQUISITIONMODE, 56
- NUM_ACQUISITIONSTATUSSELECTOR, 57
- NUM_ACTIONUNCONDITIONALMODE, 57
- NUM_ADCBITDEPTH, 57
- NUM_AUTOALGORITHMSELECTOR, 58
- NUM_AUTOEXPOSURECONTROLPRIORITY, 58
- NUM_AUTOEXPOSURELIGHTINGMODE, 58
- NUM_AUTOEXPOSUREMETERINGMODE, 59
- NUM_AUTOEXPOSURETARGETGREYVALUEAUTO, 59
- NUM_BALANCERATIOSELECTOR, 60
- NUM_BALANCEWHITEAUTO, 60
- NUM_BALANCEWHITEAUTOPROFILE, 60
- NUM_BINNINGHORIZONTALMODE, 61
- NUM_BINNINGSELECTOR, 61
- NUM_BINNINGVERTICALMODE, 61
- NUM_BLACKLEVELAUTO, 62
- NUM_BLACKLEVELAUTOBALANCE, 62
- NUM_BLACKLEVELSELECTOR, 62
- NUM_CHUNKBLACKLEVELSELECTOR, 63
- NUM_CHUNKCOUNTERSELECTOR, 63
- NUM_CHUNKENCODERSELECTOR, 63
- NUM_CHUNKENCODERSTATUS, 63
- NUM_CHUNKEXPOSURETIMESELECTOR, 64
- NUM_CHUNKGAINSELECTOR, 64
- NUM_CHUNKIMAGECOMPONENT, 65
- NUM_CHUNKPIXELFORMAT, 65
- NUM_CHUNKREGIONID, 65
- NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR, 66
- NUM_CHUNKSCAN3DCOORDINATESELECTOR, 66
- NUM_CHUNKSCAN3DCOORDINATESYSTEM, 66
- NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCESELECTOR, 67
- NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR, 67
- NUM_CHUNKSCAN3DDISTANCEUNIT, 67
- NUM_CHUNKSCAN3DOUTPUTMODE, 68
- NUM_CHUNKSELECTOR, 69
- NUM_CHUNKSOURCEID, 69
- NUM_CHUNKTIMERSELECTOR, 70
- NUM_CHUNKTRANSFERSTREAMID, 70
- NUM_CLCONFIGURATION, 70
- NUM_CLTIMESLOTSCOUNT, 71
- NUM_COLORTRANSFORMATIONSELECTOR, 71
- NUM_COLORTRANSFORMATIONVALUESELECTOR, 71
- NUM_COMPRESSIONSATURATIONPRIORITY, 72
- NUM_COUNTEREVENTACTIVATION, 72
- NUM_COUNTEREVENTSOURCE, 73
- NUM_COUNTERRESETACTIVATION, 73
- NUM_COUNTERRESETSOURCE, 74
- NUM_COUNTERSELECTOR, 74
- NUM_COUNTERSTATUS, 74
- NUM_COUNTERTRIGGERACTIVATION, 75
- NUM_COUNTERTRIGGERSOURCE, 75
- NUM_CXPCONNECTIONTESTMODE, 76
- NUM_CXPLINKCONFIGURATION, 77
- NUM_CXPLINKCONFIGURATIONPREFERRED, 78
- NUM_CXPLINKCONFIGURATIONSTATUS, 79
- NUM_CXPPOCXPSTATUS, 79
- NUM_DECIMATIONHORIZONTALMODE, 79
- NUM_DECIMATIONSELECTOR, 80
- NUM_DECIMATIONVERTICALMODE, 80
- NUM_DEFECTCORRECTIONMODE, 80
- NUM_DEINTERLACING, 81
- NUM_DEVICECHARACTERSET, 81
- NUM_DEVICECLOCKSELECTOR, 81
- NUM_DEVICECONNECTIONSTATUS, 81
- NUM_DEVICEINDICATORMODE, 82
- NUM_DEVICELINKHEARTBEATMODE, 82
- NUM_DEVICELINKTHROUGHPUTLIMITMODE, 82
- NUM_DEVICEPOWERSUPPLYSELECTOR, 83
- NUM_DEVICEREGISTERSENDIANNESS, 83
- NUM_DEVICESCANTYPE, 83
- NUM_DEVICESERIALPORTBAUDRATE, 84
- NUM_DEVICESERIALPORTSELECTOR, 84
- NUM_DEVICESTREAMCHANNELENDIANNESS, 84
- NUM_DEVICESTREAMCHANNELTYPE, 85
- NUM_DEVICETAPGEOMETRY, 86
- NUM_DEVICETEMPERATURESELECTOR, 86
- NUM_DEVICETLTYPE, 86
- NUM_DEVICETYPE, 87
- NUM_ENCODERMODE, 87
- NUM_ENCODEROUTPUTMODE, 88
- NUM_ENCODERRESETACTIVATION, 88
- NUM_ENCODERRESETSOURCE, 89
- NUM_ENCODERSELECTOR, 89
- NUM_ENCODERSOURCEA, 90
- NUM_ENCODERSOURCEB, 90
- NUM_ENCODERSTATUS, 90
- NUM_EVENTNOTIFICATION, 91
- NUM_EVENTSELECTOR, 91
- NUM_EXPOSUREACTIVEMODE, 91
- NUM_EXPOSUREAUTO, 92
- NUM_EXPOSUREMODE, 92
- NUM_EXPOSURETIMEMODE, 92

- NUM_EXPOSURETIMESELECTOR, 93
- NUM_FILEOPENMODE, 93
- NUM_FILEOPERATIONSELECTOR, 94
- NUM_FILEOPERATIONSTATUS, 94
- NUM_FILESELECTOR, 94
- NUM_GAINAUTO, 96
- NUM_GAINAUTOBALANCE, 96
- NUM_GAINSELECTOR, 96
- NUM_GEVCCP, 97
- NUM_GEVCURRENTPHYSICALLINKCONFIGURATION, 97
- NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR, 97
- NUM_GEVGVSPEXTENDEDIDMODE, 98
- NUM_GEVIEEE1588CLOCKACCURACY, 98
- NUM_GEVIEEE1588MODE, 98
- NUM_GEVIEEE1588STATUS, 99
- NUM_GEVIPCONFIGURATIONSTATUS, 99
- NUM_GEVPHYSCALLINKCONFIGURATION, 99
- NUM_GEVSUPPORTEDOPTIONSELECTOR, 100
- NUM_IMAGECOMPONENTSELECTOR, 101
- NUM_IMAGECOMPRESSIONJPEGFORMATOPTION, 101
- NUM_IMAGECOMPRESSIONMODE, 102
- NUM_IMAGECOMPRESSIONRATEOPTION, 102
- NUM_LINEFORMAT, 102
- NUM_LINEINPUTFILTERSELECTOR, 103
- NUM_LINEMODE, 103
- NUM_LINESELECTOR, 103
- NUM_LINESOURCE, 104
- NUM_LOGICBLOCKLUTINPUTACTIVATION, 104
- NUM_LOGICBLOCKLUTINPUTSELECTOR, 105
- NUM_LOGICBLOCKLUTINPUTSOURCE, 105
- NUM_LOGICBLOCKLUTSELECTOR, 106
- NUM_LOGICBLOCKSELECTOR, 106
- NUM_LUTSELECTOR, 106
- NUM_PIXELCOLORFILTER, 107
- NUM_PIXELFORMAT, 112
- NUM_PIXELFORMATINFOSELECTOR, 118
- NUM_PIXELSIZE, 119
- NUM_REGIONDESTINATION, 119
- NUM_REGIONMODE, 119
- NUM_REGIONSELECTOR, 120
- NUM_RGBTRANSFORMLIGHTSOURCE, 120
- NUM_SCAN3DCOORDINATEREFERENCESELECTOR, 121
- NUM_SCAN3DCOORDINATESELECTOR, 121
- NUM_SCAN3DCOORDINATESYSTEM, 121
- NUM_SCAN3DCOORDINATESYSTEMREFERENCE, 122
- NUM_SCAN3DCOORDINATETRANSFORMSELECTOR, 122
- NUM_SCAN3DDISTANCEUNIT, 122
- NUM_SCAN3DOUTPUTMODE, 124
- NUM_SENSORDIGITIZATIONTAPS, 125
- NUM_SENSORSHUTTERMODE, 125
- NUM_SENSORTAPS, 125
- NUM_SEQUENCERCONFIGURATIONMODE, 126
- NUM_SEQUENCERCONFIGURATIONVALID, 126
- NUM_SEQUENCERMODE, 126
- NUM_SEQUENCERSETVALID, 127
- NUM_SEQUENCERTRIGGERACTIVATION, 127
- NUM_SEQUENCERTRIGGERSOURCE, 127
- NUM_SERIALPORTBAUDRATE, 128
- NUM_SERIALPORTPARITY, 128
- NUM_SERIALPORTSELECTOR, 128
- NUM_SERIALPORTSOURCE, 129
- NUM_SERIALPORTSTOPBITS, 129
- NUM_SOFTWARESIGNALSELECTOR, 129
- NUM_SOURCESELECTOR, 130
- NUM_TESTPATTERN, 130
- NUM_TESTPATTERNGENERATORSELECTOR, 130
- NUM_TIMERSELECTOR, 131
- NUM_TIMERSTATUS, 131
- NUM_TIMERTRIGGERACTIVATION, 131
- NUM_TIMERTRIGGERSOURCE, 133
- NUM_TRANSFERCOMPONENTSELECTOR, 133
- NUM_TRANSFERCONTROLMODE, 134
- NUM_TRANSFEROPERATIONMODE, 134
- NUM_TRANSFERQUEUEMODE, 134
- NUM_TRANSFERSELECTOR, 134
- NUM_TRANSFERSTATUSSELECTOR, 135
- NUM_TRANSFERTRIGGERACTIVATION, 135
- NUM_TRANSFERTRIGGERMODE, 136
- NUM_TRANSFERTRIGGERSELECTOR, 136
- NUM_TRANSFERTRIGGERSOURCE, 137
- NUM_TRIGGERACTIVATION, 137
- NUM_TRIGGERMODE, 138
- NUM_TRIGGEROVERLAP, 138
- NUM_TRIGGERSELECTOR, 138
- NUM_TRIGGERSOURCE, 139
- NUM_USEROUTPUTSELECTOR, 139
- NUM_USERSETDEFAULT, 140
- NUM_USERSETSELECTOR, 140
- NUM_WHITECLIPSELECTOR, 140
- PixelColorFilter_BayerBG, 107
- PixelColorFilter_BayerGB, 107
- PixelColorFilter_BayerGR, 107
- PixelColorFilter_BayerRG, 106
- PixelColorFilter_None, 106
- PixelFormat_B10, 109
- PixelFormat_B12, 109
- PixelFormat_B12_Jpeg, 112
- PixelFormat_B16, 109
- PixelFormat_B8, 109
- PixelFormat_BayerBG10, 108
- PixelFormat_BayerBG10p, 108
- PixelFormat_BayerBG10Packed, 108
- PixelFormat_BayerBG12, 108
- PixelFormat_BayerBG12p, 107
- PixelFormat_BayerBG12Packed, 107
- PixelFormat_BayerBG16, 107
- PixelFormat_BayerBG8, 107

PixelFormat_BayerGB10, [108](#)
PixelFormat_BayerGB10p, [108](#)
PixelFormat_BayerGB10Packed, [108](#)
PixelFormat_BayerGB12, [108](#)
PixelFormat_BayerGB12p, [107](#)
PixelFormat_BayerGB12Packed, [107](#)
PixelFormat_BayerGB16, [107](#)
PixelFormat_BayerGB8, [107](#)
PixelFormat_BayerGR10, [108](#)
PixelFormat_BayerGR10p, [108](#)
PixelFormat_BayerGR10Packed, [107](#)
PixelFormat_BayerGR12, [108](#)
PixelFormat_BayerGR12p, [107](#)
PixelFormat_BayerGR12Packed, [107](#)
PixelFormat_BayerGR16, [107](#)
PixelFormat_BayerGR8, [107](#)
PixelFormat_BayerRG10, [108](#)
PixelFormat_BayerRG10p, [108](#)
PixelFormat_BayerRG10Packed, [107](#)
PixelFormat_BayerRG12, [108](#)
PixelFormat_BayerRG12p, [107](#)
PixelFormat_BayerRG12Packed, [107](#)
PixelFormat_BayerRG16, [107](#)
PixelFormat_BayerRG8, [107](#)
PixelFormat_BayerRGPolarized10p, [112](#)
PixelFormat_BayerRGPolarized12p, [112](#)
PixelFormat_BayerRGPolarized16, [112](#)
PixelFormat_BayerRGPolarized8, [112](#)
PixelFormat_BGR10, [109](#)
PixelFormat_BGR10p, [109](#)
PixelFormat_BGR12, [109](#)
PixelFormat_BGR12p, [109](#)
PixelFormat_BGR14, [109](#)
PixelFormat_BGR16, [109](#)
PixelFormat_BGR565p, [109](#)
PixelFormat_BGR8, [107](#)
PixelFormat_BGRa10, [109](#)
PixelFormat_BGRa10p, [109](#)
PixelFormat_BGRa12, [109](#)
PixelFormat_BGRa12p, [109](#)
PixelFormat_BGRa14, [109](#)
PixelFormat_BGRa16, [109](#)
PixelFormat_BGRa8, [107](#)
PixelFormat_BiColorBGRG10, [110](#)
PixelFormat_BiColorBGRG10p, [110](#)
PixelFormat_BiColorBGRG12, [110](#)
PixelFormat_BiColorBGRG12p, [110](#)
PixelFormat_BiColorBGRG8, [110](#)
PixelFormat_BiColorRGBG10, [110](#)
PixelFormat_BiColorRGBG10p, [110](#)
PixelFormat_BiColorRGBG12, [110](#)
PixelFormat_BiColorRGBG12p, [110](#)
PixelFormat_BiColorRGBG8, [110](#)
PixelFormat_Confidence1, [110](#)
PixelFormat_Confidence16, [110](#)
PixelFormat_Confidence1p, [110](#)
PixelFormat_Confidence32f, [110](#)
PixelFormat_Confidence8, [110](#)
PixelFormat_Coord3D_A10p, [110](#)
PixelFormat_Coord3D_A12p, [110](#)
PixelFormat_Coord3D_A16, [110](#)
PixelFormat_Coord3D_A32f, [110](#)
PixelFormat_Coord3D_A8, [110](#)
PixelFormat_Coord3D_ABC10p, [109](#)
PixelFormat_Coord3D_ABC10p_Planar, [109](#)
PixelFormat_Coord3D_ABC12p, [109](#)
PixelFormat_Coord3D_ABC12p_Planar, [109](#)
PixelFormat_Coord3D_ABC16, [109](#)
PixelFormat_Coord3D_ABC16_Planar, [109](#)
PixelFormat_Coord3D_ABC32f, [109](#)
PixelFormat_Coord3D_ABC32f_Planar, [109](#)
PixelFormat_Coord3D_ABC8, [109](#)
PixelFormat_Coord3D_ABC8_Planar, [109](#)
PixelFormat_Coord3D_AC10p, [109](#)
PixelFormat_Coord3D_AC10p_Planar, [109](#)
PixelFormat_Coord3D_AC12p, [109](#)
PixelFormat_Coord3D_AC12p_Planar, [109](#)
PixelFormat_Coord3D_AC16, [109](#)
PixelFormat_Coord3D_AC16_Planar, [109](#)
PixelFormat_Coord3D_AC32f, [109](#)
PixelFormat_Coord3D_AC32f_Planar, [109](#)
PixelFormat_Coord3D_AC8, [109](#)
PixelFormat_Coord3D_AC8_Planar, [109](#)
PixelFormat_Coord3D_B10p, [110](#)
PixelFormat_Coord3D_B12p, [110](#)
PixelFormat_Coord3D_B16, [110](#)
PixelFormat_Coord3D_B32f, [110](#)
PixelFormat_Coord3D_B8, [110](#)
PixelFormat_Coord3D_C10p, [110](#)
PixelFormat_Coord3D_C12p, [110](#)
PixelFormat_Coord3D_C16, [110](#)
PixelFormat_Coord3D_C32f, [110](#)
PixelFormat_Coord3D_C8, [110](#)
PixelFormat_G10, [109](#)
PixelFormat_G12, [109](#)
PixelFormat_G16, [109](#)
PixelFormat_G8, [109](#)
PixelFormat_GB12, [112](#)
PixelFormat_GB12_Jpeg, [112](#)
PixelFormat_GR12, [112](#)
PixelFormat_GR12_Jpeg, [112](#)
PixelFormat_JPEGColor8, [112](#)
PixelFormat_JPEGMono8, [112](#)
PixelFormat_LLCBayerRG8, [112](#)
PixelFormat_LLCMono8, [112](#)
PixelFormat_Mono10, [108](#)
PixelFormat_Mono10p, [108](#)
PixelFormat_Mono10Packed, [107](#)
PixelFormat_Mono12, [108](#)
PixelFormat_Mono12p, [107](#)
PixelFormat_Mono12Packed, [107](#)
PixelFormat_Mono14, [108](#)
PixelFormat_Mono16, [107](#)
PixelFormat_Mono16s, [108](#)
PixelFormat_Mono1p, [108](#)
PixelFormat_Mono2p, [108](#)

PixelFormat_Mono32f, [108](#)
PixelFormat_Mono4p, [108](#)
PixelFormat_Mono8, [107](#)
PixelFormat_Mono8s, [108](#)
PixelFormat_Polarized10p, [112](#)
PixelFormat_Polarized12p, [112](#)
PixelFormat_Polarized16, [112](#)
PixelFormat_Polarized8, [112](#)
PixelFormat_R10, [109](#)
PixelFormat_R12, [109](#)
PixelFormat_R12_Jpeg, [112](#)
PixelFormat_R16, [109](#)
PixelFormat_R8, [109](#)
PixelFormat_Raw16, [112](#)
PixelFormat_Raw8, [112](#)
PixelFormat_RGB10, [108](#)
PixelFormat_RGB10_Planar, [108](#)
PixelFormat_RGB10p, [108](#)
PixelFormat_RGB10p32, [108](#)
PixelFormat_RGB12, [108](#)
PixelFormat_RGB12_Planar, [108](#)
PixelFormat_RGB12p, [108](#)
PixelFormat_RGB14, [108](#)
PixelFormat_RGB16, [108](#)
PixelFormat_RGB16_Planar, [108](#)
PixelFormat_RGB16s, [108](#)
PixelFormat_RGB32f, [108](#)
PixelFormat_RGB565p, [109](#)
PixelFormat_RGB8, [108](#)
PixelFormat_RGB8_Planar, [108](#)
PixelFormat_RGB8Packed, [107](#)
PixelFormat_RGBa10, [108](#)
PixelFormat_RGBa10p, [108](#)
PixelFormat_RGBa12, [108](#)
PixelFormat_RGBa12p, [108](#)
PixelFormat_RGBa14, [108](#)
PixelFormat_RGBa16, [108](#)
PixelFormat_RGBa32f, [109](#)
PixelFormat_RGBa8, [108](#)
PixelFormat_SCF1WBWG10, [110](#)
PixelFormat_SCF1WBWG10p, [110](#)
PixelFormat_SCF1WBWG12, [110](#)
PixelFormat_SCF1WBWG12p, [110](#)
PixelFormat_SCF1WBWG14, [110](#)
PixelFormat_SCF1WBWG16, [110](#)
PixelFormat_SCF1WBWG8, [110](#)
PixelFormat_SCF1WGWB10, [110](#)
PixelFormat_SCF1WGWB10p, [110](#)
PixelFormat_SCF1WGWB12, [110](#)
PixelFormat_SCF1WGWB12p, [110](#)
PixelFormat_SCF1WGWB14, [110](#)
PixelFormat_SCF1WGWB16, [110](#)
PixelFormat_SCF1WGWB8, [110](#)
PixelFormat_SCF1WGWR10, [110](#)
PixelFormat_SCF1WGWR10p, [110](#)
PixelFormat_SCF1WGWR12, [111](#)
PixelFormat_SCF1WGWR12p, [111](#)
PixelFormat_SCF1WGWR14, [111](#)
PixelFormat_SCF1WGWR16, [111](#)
PixelFormat_SCF1WGWR8, [110](#)
PixelFormat_SCF1WRWG10, [111](#)
PixelFormat_SCF1WRWG10p, [111](#)
PixelFormat_SCF1WRWG12, [111](#)
PixelFormat_SCF1WRWG12p, [111](#)
PixelFormat_SCF1WRWG14, [111](#)
PixelFormat_SCF1WRWG16, [111](#)
PixelFormat_SCF1WRWG8, [111](#)
PixelFormat_YCbCr10_CbYCr, [111](#)
PixelFormat_YCbCr10p_CbYCr, [111](#)
PixelFormat_YCbCr12_CbYCr, [111](#)
PixelFormat_YCbCr12p_CbYCr, [111](#)
PixelFormat_YCbCr411_8, [107](#)
PixelFormat_YCbCr411_8_CbYYCrYY, [111](#)
PixelFormat_YCbCr422_10, [111](#)
PixelFormat_YCbCr422_10_CbYCrY, [111](#)
PixelFormat_YCbCr422_10p, [111](#)
PixelFormat_YCbCr422_10p_CbYCrY, [111](#)
PixelFormat_YCbCr422_12, [111](#)
PixelFormat_YCbCr422_12_CbYCrY, [111](#)
PixelFormat_YCbCr422_12p, [111](#)
PixelFormat_YCbCr422_12p_CbYCrY, [111](#)
PixelFormat_YCbCr422_8, [107](#)
PixelFormat_YCbCr422_8_CbYCrY, [111](#)
PixelFormat_YCbCr601_10_CbYCr, [111](#)
PixelFormat_YCbCr601_10p_CbYCr, [111](#)
PixelFormat_YCbCr601_12_CbYCr, [111](#)
PixelFormat_YCbCr601_12p_CbYCr, [111](#)
PixelFormat_YCbCr601_411_8_CbYYCrYY, [111](#)
PixelFormat_YCbCr601_422_10, [111](#)
PixelFormat_YCbCr601_422_10_CbYCrY, [111](#)
PixelFormat_YCbCr601_422_10p, [111](#)
PixelFormat_YCbCr601_422_10p_CbYCrY, [111](#)
PixelFormat_YCbCr601_422_12, [111](#)
PixelFormat_YCbCr601_422_12_CbYCrY, [111](#)
PixelFormat_YCbCr601_422_12p, [111](#)
PixelFormat_YCbCr601_422_12p_CbYCrY, [111](#)
PixelFormat_YCbCr601_422_8, [111](#)
PixelFormat_YCbCr601_422_8_CbYCrY, [111](#)
PixelFormat_YCbCr601_8_CbYCr, [111](#)
PixelFormat_YCbCr709_10_CbYCr, [111](#)
PixelFormat_YCbCr709_10p_CbYCr, [111](#)
PixelFormat_YCbCr709_12_CbYCr, [111](#)
PixelFormat_YCbCr709_12p_CbYCr, [111](#)
PixelFormat_YCbCr709_411_8_CbYYCrYY, [112](#)
PixelFormat_YCbCr709_422_10, [112](#)
PixelFormat_YCbCr709_422_10_CbYCrY, [112](#)
PixelFormat_YCbCr709_422_10p, [112](#)
PixelFormat_YCbCr709_422_10p_CbYCrY, [112](#)
PixelFormat_YCbCr709_422_12, [112](#)
PixelFormat_YCbCr709_422_12_CbYCrY, [112](#)
PixelFormat_YCbCr709_422_12p, [112](#)
PixelFormat_YCbCr709_422_12p_CbYCrY, [112](#)
PixelFormat_YCbCr709_422_8, [112](#)
PixelFormat_YCbCr709_422_8_CbYCrY, [112](#)
PixelFormat_YCbCr709_8_CbYCr, [111](#)
PixelFormat_YCbCr8, [107](#)

- PixelFormat_YCbCr8_CbYCr, [111](#)
- PixelFormat_YUV411_8_UYYVYY, [112](#)
- PixelFormat_YUV411Packed, [107](#)
- PixelFormat_YUV422_8, [112](#)
- PixelFormat_YUV422_8_UYVY, [112](#)
- PixelFormat_YUV422Packed, [107](#)
- PixelFormat_YUV444Packed, [107](#)
- PixelFormat_YUV8_UYV, [112](#)
- PixelFormatInfoSelector_B10, [114](#)
- PixelFormatInfoSelector_B12, [114](#)
- PixelFormatInfoSelector_B16, [114](#)
- PixelFormatInfoSelector_B8, [114](#)
- PixelFormatInfoSelector_BayerBG10, [113](#)
- PixelFormatInfoSelector_BayerBG10p, [113](#)
- PixelFormatInfoSelector_BayerBG12, [113](#)
- PixelFormatInfoSelector_BayerBG12p, [113](#)
- PixelFormatInfoSelector_BayerBG16, [113](#)
- PixelFormatInfoSelector_BayerBG8, [113](#)
- PixelFormatInfoSelector_BayerGB10, [113](#)
- PixelFormatInfoSelector_BayerGB10p, [113](#)
- PixelFormatInfoSelector_BayerGB12, [113](#)
- PixelFormatInfoSelector_BayerGB12p, [113](#)
- PixelFormatInfoSelector_BayerGB16, [113](#)
- PixelFormatInfoSelector_BayerGB8, [113](#)
- PixelFormatInfoSelector_BayerGR10, [113](#)
- PixelFormatInfoSelector_BayerGR10p, [113](#)
- PixelFormatInfoSelector_BayerGR12, [113](#)
- PixelFormatInfoSelector_BayerGR12p, [113](#)
- PixelFormatInfoSelector_BayerGR16, [113](#)
- PixelFormatInfoSelector_BayerGR8, [113](#)
- PixelFormatInfoSelector_BayerRG10, [113](#)
- PixelFormatInfoSelector_BayerRG10p, [113](#)
- PixelFormatInfoSelector_BayerRG12, [113](#)
- PixelFormatInfoSelector_BayerRG12p, [113](#)
- PixelFormatInfoSelector_BayerRG16, [113](#)
- PixelFormatInfoSelector_BayerRG8, [113](#)
- PixelFormatInfoSelector_BayerRGPolarized10p, [118](#)
- PixelFormatInfoSelector_BayerRGPolarized12p, [118](#)
- PixelFormatInfoSelector_BayerRGPolarized16, [118](#)
- PixelFormatInfoSelector_BayerRGPolarized8, [118](#)
- PixelFormatInfoSelector_BGR10, [114](#)
- PixelFormatInfoSelector_BGR10p, [114](#)
- PixelFormatInfoSelector_BGR12, [114](#)
- PixelFormatInfoSelector_BGR12p, [114](#)
- PixelFormatInfoSelector_BGR14, [114](#)
- PixelFormatInfoSelector_BGR16, [114](#)
- PixelFormatInfoSelector_BGR565p, [114](#)
- PixelFormatInfoSelector_BGR8, [114](#)
- PixelFormatInfoSelector_BGRa10, [114](#)
- PixelFormatInfoSelector_BGRa10p, [114](#)
- PixelFormatInfoSelector_BGRa12, [114](#)
- PixelFormatInfoSelector_BGRa12p, [114](#)
- PixelFormatInfoSelector_BGRa14, [114](#)
- PixelFormatInfoSelector_BGRa16, [114](#)
- PixelFormatInfoSelector_BGRa8, [114](#)
- PixelFormatInfoSelector_BiColorBGRG10, [115](#)
- PixelFormatInfoSelector_BiColorBGRG10p, [115](#)
- PixelFormatInfoSelector_BiColorBGRG12, [115](#)
- PixelFormatInfoSelector_BiColorBGRG12p, [115](#)
- PixelFormatInfoSelector_BiColorBGRG8, [115](#)
- PixelFormatInfoSelector_BiColorRGBG10, [115](#)
- PixelFormatInfoSelector_BiColorRGBG10p, [115](#)
- PixelFormatInfoSelector_BiColorRGBG12, [115](#)
- PixelFormatInfoSelector_BiColorRGBG12p, [115](#)
- PixelFormatInfoSelector_BiColorRGBG8, [115](#)
- PixelFormatInfoSelector_Confidence1, [115](#)
- PixelFormatInfoSelector_Confidence16, [115](#)
- PixelFormatInfoSelector_Confidence1p, [115](#)
- PixelFormatInfoSelector_Confidence32f, [115](#)
- PixelFormatInfoSelector_Confidence8, [115](#)
- PixelFormatInfoSelector_Coord3D_A10p, [115](#)
- PixelFormatInfoSelector_Coord3D_A12p, [115](#)
- PixelFormatInfoSelector_Coord3D_A16, [115](#)
- PixelFormatInfoSelector_Coord3D_A32f, [115](#)
- PixelFormatInfoSelector_Coord3D_A8, [115](#)
- PixelFormatInfoSelector_Coord3D_ABC10p, [114](#)
- PixelFormatInfoSelector_Coord3D_ABC10p_Planar, [114](#)
- PixelFormatInfoSelector_Coord3D_ABC12p, [114](#)
- PixelFormatInfoSelector_Coord3D_ABC12p_Planar, [114](#)
- PixelFormatInfoSelector_Coord3D_ABC16, [115](#)
- PixelFormatInfoSelector_Coord3D_ABC16_Planar, [115](#)
- PixelFormatInfoSelector_Coord3D_ABC32f, [115](#)
- PixelFormatInfoSelector_Coord3D_ABC32f_Planar, [115](#)
- PixelFormatInfoSelector_Coord3D_ABC8, [114](#)
- PixelFormatInfoSelector_Coord3D_ABC8_Planar, [114](#)
- PixelFormatInfoSelector_Coord3D_AC10p, [115](#)
- PixelFormatInfoSelector_Coord3D_AC10p_Planar, [115](#)
- PixelFormatInfoSelector_Coord3D_AC12p, [115](#)
- PixelFormatInfoSelector_Coord3D_AC12p_Planar, [115](#)
- PixelFormatInfoSelector_Coord3D_AC16, [115](#)
- PixelFormatInfoSelector_Coord3D_AC16_Planar, [115](#)
- PixelFormatInfoSelector_Coord3D_AC32f, [115](#)
- PixelFormatInfoSelector_Coord3D_AC32f_Planar, [115](#)
- PixelFormatInfoSelector_Coord3D_AC8, [115](#)
- PixelFormatInfoSelector_Coord3D_AC8_Planar, [115](#)
- PixelFormatInfoSelector_Coord3D_B10p, [115](#)
- PixelFormatInfoSelector_Coord3D_B12p, [115](#)
- PixelFormatInfoSelector_Coord3D_B16, [115](#)
- PixelFormatInfoSelector_Coord3D_B32f, [115](#)
- PixelFormatInfoSelector_Coord3D_B8, [115](#)
- PixelFormatInfoSelector_Coord3D_C10p, [115](#)
- PixelFormatInfoSelector_Coord3D_C12p, [115](#)
- PixelFormatInfoSelector_Coord3D_C16, [115](#)

- PixelFormatInfoSelector_Coord3D_C32f, 115
- PixelFormatInfoSelector_Coord3D_C8, 115
- PixelFormatInfoSelector_G10, 114
- PixelFormatInfoSelector_G12, 114
- PixelFormatInfoSelector_G16, 114
- PixelFormatInfoSelector_G8, 114
- PixelFormatInfoSelector_JPEGColor8, 118
- PixelFormatInfoSelector_JPEGMono8, 118
- PixelFormatInfoSelector_LLCBayerRG8, 118
- PixelFormatInfoSelector_LLCMono8, 118
- PixelFormatInfoSelector_Mono10, 113
- PixelFormatInfoSelector_Mono10p, 113
- PixelFormatInfoSelector_Mono12, 113
- PixelFormatInfoSelector_Mono12p, 113
- PixelFormatInfoSelector_Mono14, 113
- PixelFormatInfoSelector_Mono16, 113
- PixelFormatInfoSelector_Mono16s, 113
- PixelFormatInfoSelector_Mono1p, 113
- PixelFormatInfoSelector_Mono2p, 113
- PixelFormatInfoSelector_Mono32f, 113
- PixelFormatInfoSelector_Mono4p, 113
- PixelFormatInfoSelector_Mono8, 113
- PixelFormatInfoSelector_Mono8s, 113
- PixelFormatInfoSelector_Polarized10p, 118
- PixelFormatInfoSelector_Polarized12p, 118
- PixelFormatInfoSelector_Polarized16, 118
- PixelFormatInfoSelector_Polarized8, 118
- PixelFormatInfoSelector_R10, 114
- PixelFormatInfoSelector_R12, 114
- PixelFormatInfoSelector_R16, 114
- PixelFormatInfoSelector_R8, 114
- PixelFormatInfoSelector_RGB10, 113
- PixelFormatInfoSelector_RGB10_Planar, 114
- PixelFormatInfoSelector_RGB10p, 114
- PixelFormatInfoSelector_RGB10p32, 114
- PixelFormatInfoSelector_RGB12, 114
- PixelFormatInfoSelector_RGB12_Planar, 114
- PixelFormatInfoSelector_RGB12p, 114
- PixelFormatInfoSelector_RGB14, 114
- PixelFormatInfoSelector_RGB16, 114
- PixelFormatInfoSelector_RGB16_Planar, 114
- PixelFormatInfoSelector_RGB16s, 114
- PixelFormatInfoSelector_RGB32f, 114
- PixelFormatInfoSelector_RGB565p, 114
- PixelFormatInfoSelector_RGB8, 113
- PixelFormatInfoSelector_RGB8_Planar, 113
- PixelFormatInfoSelector_RGBa10, 113
- PixelFormatInfoSelector_RGBa10p, 113
- PixelFormatInfoSelector_RGBa12, 113
- PixelFormatInfoSelector_RGBa12p, 113
- PixelFormatInfoSelector_RGBa14, 113
- PixelFormatInfoSelector_RGBa16, 113
- PixelFormatInfoSelector_RGBa32f, 114
- PixelFormatInfoSelector_RGBa8, 113
- PixelFormatInfoSelector_SCF1WBWG10, 115
- PixelFormatInfoSelector_SCF1WBWG10p, 116
- PixelFormatInfoSelector_SCF1WBWG12, 116
- PixelFormatInfoSelector_SCF1WBWG12p, 116
- PixelFormatInfoSelector_SCF1WBWG14, 116
- PixelFormatInfoSelector_SCF1WBWG16, 116
- PixelFormatInfoSelector_SCF1WBWG8, 115
- PixelFormatInfoSelector_SCF1WGWB10, 116
- PixelFormatInfoSelector_SCF1WGWB10p, 116
- PixelFormatInfoSelector_SCF1WGWB12, 116
- PixelFormatInfoSelector_SCF1WGWB12p, 116
- PixelFormatInfoSelector_SCF1WGWB14, 116
- PixelFormatInfoSelector_SCF1WGWB16, 116
- PixelFormatInfoSelector_SCF1WGWB8, 116
- PixelFormatInfoSelector_SCF1WGWR10, 116
- PixelFormatInfoSelector_SCF1WGWR10p, 116
- PixelFormatInfoSelector_SCF1WGWR12, 116
- PixelFormatInfoSelector_SCF1WGWR12p, 116
- PixelFormatInfoSelector_SCF1WGWR14, 116
- PixelFormatInfoSelector_SCF1WGWR16, 116
- PixelFormatInfoSelector_SCF1WGWR8, 116
- PixelFormatInfoSelector_SCF1WRWG10, 116
- PixelFormatInfoSelector_SCF1WRWG10p, 116
- PixelFormatInfoSelector_SCF1WRWG12, 116
- PixelFormatInfoSelector_SCF1WRWG12p, 116
- PixelFormatInfoSelector_SCF1WRWG14, 116
- PixelFormatInfoSelector_SCF1WRWG16, 116
- PixelFormatInfoSelector_SCF1WRWG8, 116
- PixelFormatInfoSelector_YCbCr10_CbYCr, 116
- PixelFormatInfoSelector_YCbCr10p_CbYCr, 116
- PixelFormatInfoSelector_YCbCr12_CbYCr, 117
- PixelFormatInfoSelector_YCbCr12p_CbYCr, 117
- PixelFormatInfoSelector_YCbCr411_8, 117
- PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY, 117
- PixelFormatInfoSelector_YCbCr422_10, 117
- PixelFormatInfoSelector_YCbCr422_10_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr422_10p, 117
- PixelFormatInfoSelector_YCbCr422_10p_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr422_12, 117
- PixelFormatInfoSelector_YCbCr422_12_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr422_12p, 117
- PixelFormatInfoSelector_YCbCr422_12p_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr422_8, 117
- PixelFormatInfoSelector_YCbCr422_8_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr601_10_CbYCr, 117
- PixelFormatInfoSelector_YCbCr601_10p_CbYCr, 117
- PixelFormatInfoSelector_YCbCr601_12_CbYCr, 117
- PixelFormatInfoSelector_YCbCr601_12p_CbYCr, 117
- PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY, 117
- PixelFormatInfoSelector_YCbCr601_422_10, 117
- PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY, 117

- 117
- PixelFormatInfoSelector_YCbCr601_422_10p, 117
- PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr601_422_12, 117
- PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr601_422_12p, 117
- PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr601_422_8, 117
- PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr601_8_CbYCr, 117
- PixelFormatInfoSelector_YCbCr709_10_CbYCr, 117
- PixelFormatInfoSelector_YCbCr709_10p_CbYCr, 117
- PixelFormatInfoSelector_YCbCr709_12_CbYCr, 117
- PixelFormatInfoSelector_YCbCr709_12p_CbYCr, 117
- PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY, 117
- PixelFormatInfoSelector_YCbCr709_422_10, 117
- PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr709_422_10p, 117
- PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY, 118
- PixelFormatInfoSelector_YCbCr709_422_12, 118
- PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY, 118
- PixelFormatInfoSelector_YCbCr709_422_12p, 118
- PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY, 118
- PixelFormatInfoSelector_YCbCr709_422_8, 117
- PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY, 117
- PixelFormatInfoSelector_YCbCr709_8_CbYCr, 117
- PixelFormatInfoSelector_YCbCr8, 116
- PixelFormatInfoSelector_YCbCr8_CbYCr, 116
- PixelFormatInfoSelector_YUV411_8_UYYVYY, 118
- PixelFormatInfoSelector_YUV422_8, 118
- PixelFormatInfoSelector_YUV422_8_UYVY, 118
- PixelFormatInfoSelector_YUV8_UYV, 118
- PixelSize_Bpp1, 118
- PixelSize_Bpp10, 118
- PixelSize_Bpp12, 118
- PixelSize_Bpp14, 118
- PixelSize_Bpp16, 118
- PixelSize_Bpp2, 118
- PixelSize_Bpp20, 118
- PixelSize_Bpp24, 118
- PixelSize_Bpp30, 118
- PixelSize_Bpp32, 118
- PixelSize_Bpp36, 119
- PixelSize_Bpp4, 118
- PixelSize_Bpp48, 119
- PixelSize_Bpp64, 119
- PixelSize_Bpp8, 118
- PixelSize_Bpp96, 119
- RegionDestination_Stream0, 119
- RegionDestination_Stream1, 119
- RegionDestination_Stream2, 119
- RegionMode_Off, 119
- RegionMode_On, 119
- RegionSelector_All, 120
- RegionSelector_Region0, 120
- RegionSelector_Region1, 120
- RegionSelector_Region2, 120
- RgbTransformLightSource_Cloudy6500K, 120
- RgbTransformLightSource_CoolFluorescent4000K, 120
- RgbTransformLightSource_Custom, 120
- RgbTransformLightSource_Daylight5000K, 120
- RgbTransformLightSource_General, 120
- RgbTransformLightSource_Shade8000K, 120
- RgbTransformLightSource_Tungsten2800K, 120
- RgbTransformLightSource_WarmFluorescent3000K, 120
- Scan3dCoordinateReferenceSelector_RotationX, 121
- Scan3dCoordinateReferenceSelector_RotationY, 121
- Scan3dCoordinateReferenceSelector_RotationZ, 121
- Scan3dCoordinateReferenceSelector_TranslationX, 121
- Scan3dCoordinateReferenceSelector_TranslationY, 121
- Scan3dCoordinateReferenceSelector_TranslationZ, 121
- Scan3dCoordinateSelector_CoordinateA, 121
- Scan3dCoordinateSelector_CoordinateB, 121
- Scan3dCoordinateSelector_CoordinateC, 121
- Scan3dCoordinateSystem_Cartesian, 121
- Scan3dCoordinateSystem_Cylindrical, 121
- Scan3dCoordinateSystem_Spherical, 121
- Scan3dCoordinateSystemReference_Anchor, 122
- Scan3dCoordinateSystemReference_Transformed, 122
- Scan3dCoordinateTransformSelector_RotationX, 122
- Scan3dCoordinateTransformSelector_RotationY, 122
- Scan3dCoordinateTransformSelector_RotationZ, 122
- Scan3dCoordinateTransformSelector_TranslationX, 122
- Scan3dCoordinateTransformSelector_TranslationY, 122
- Scan3dCoordinateTransformSelector_TranslationZ, 122

- 122
- Scan3dDistanceUnit_Inch, 122
- Scan3dDistanceUnit_Millimeter, 122
- Scan3dOutputMode_CalibratedABC_Grid, 124
- Scan3dOutputMode_CalibratedABC_PointCloud, 124
- Scan3dOutputMode_CalibratedAC, 124
- Scan3dOutputMode_CalibratedAC_Linescan, 124
- Scan3dOutputMode_CalibratedC, 124
- Scan3dOutputMode_CalibratedC_Linescan, 124
- Scan3dOutputMode_DisparityC, 124
- Scan3dOutputMode_DisparityC_Linescan, 124
- Scan3dOutputMode_RectifiedC, 124
- Scan3dOutputMode_RectifiedC_Linescan, 124
- Scan3dOutputMode_UncalibratedC, 124
- SensorDigitizationTaps_Eight, 125
- SensorDigitizationTaps_Four, 125
- SensorDigitizationTaps_One, 124
- SensorDigitizationTaps_Ten, 125
- SensorDigitizationTaps_Three, 124
- SensorDigitizationTaps_Two, 124
- SensorShutterMode_Global, 125
- SensorShutterMode_GlobalReset, 125
- SensorShutterMode_Rolling, 125
- SensorTaps_Eight, 125
- SensorTaps_Four, 125
- SensorTaps_One, 125
- SensorTaps_Ten, 125
- SensorTaps_Three, 125
- SensorTaps_Two, 125
- SequencerConfigurationMode_Off, 126
- SequencerConfigurationMode_On, 126
- SequencerConfigurationValid_No, 126
- SequencerConfigurationValid_Yes, 126
- SequencerMode_Off, 126
- SequencerMode_On, 126
- SequencerSetValid_No, 127
- SequencerSetValid_Yes, 127
- SequencerTriggerActivation_AnyEdge, 127
- SequencerTriggerActivation_FallingEdge, 127
- SequencerTriggerActivation_LevelHigh, 127
- SequencerTriggerActivation_LevelLow, 127
- SequencerTriggerActivation_RisingEdge, 127
- SequencerTriggerSource_FrameStart, 127
- SequencerTriggerSource_Off, 127
- SerialPortBaudRate_Baud115200, 128
- SerialPortBaudRate_Baud1200, 128
- SerialPortBaudRate_Baud14400, 128
- SerialPortBaudRate_Baud19200, 128
- SerialPortBaudRate_Baud230400, 128
- SerialPortBaudRate_Baud2400, 128
- SerialPortBaudRate_Baud300, 128
- SerialPortBaudRate_Baud38400, 128
- SerialPortBaudRate_Baud460800, 128
- SerialPortBaudRate_Baud4800, 128
- SerialPortBaudRate_Baud57600, 128
- SerialPortBaudRate_Baud600, 128
- SerialPortBaudRate_Baud921600, 128
- SerialPortBaudRate_Baud9600, 128
- SerialPortParity_Even, 128
- SerialPortParity_Mark, 128
- SerialPortParity_None, 128
- SerialPortParity_Odd, 128
- SerialPortParity_Space, 128
- SerialPortSelector_SerialPort0, 128
- SerialPortSource_Line0, 129
- SerialPortSource_Line1, 129
- SerialPortSource_Line2, 129
- SerialPortSource_Line3, 129
- SerialPortSource_Off, 129
- SerialPortStopBits_Bits1, 129
- SerialPortStopBits_Bits1AndAHalf, 129
- SerialPortStopBits_Bits2, 129
- SoftwareSignalSelector_SoftwareSignal0, 129
- SoftwareSignalSelector_SoftwareSignal1, 129
- SoftwareSignalSelector_SoftwareSignal2, 129
- SourceSelector_All, 130
- SourceSelector_Source0, 130
- SourceSelector_Source1, 130
- SourceSelector_Source2, 130
- spinAcquisitionModeEnums, 56
- spinAcquisitionStatusSelectorEnums, 56
- spinActionUnconditionalModeEnums, 57
- spinAdcBitDepthEnums, 57
- spinAutoAlgorithmSelectorEnums, 57
- spinAutoExposureControlPriorityEnums, 58
- spinAutoExposureLightingModeEnums, 58
- spinAutoExposureMeteringModeEnums, 59
- spinAutoExposureTargetGreyValueAutoEnums, 59
- spinBalanceRatioSelectorEnums, 59
- spinBalanceWhiteAutoEnums, 60
- spinBalanceWhiteAutoProfileEnums, 60
- spinBinningHorizontalModeEnums, 60
- spinBinningSelectorEnums, 61
- spinBinningVerticalModeEnums, 61
- spinBlackLevelAutoBalanceEnums, 61
- spinBlackLevelAutoEnums, 62
- spinBlackLevelSelectorEnums, 62
- spinChunkBlackLevelSelectorEnums, 62
- spinChunkCounterSelectorEnums, 63
- spinChunkEncoderSelectorEnums, 63
- spinChunkEncoderStatusEnums, 63
- spinChunkExposureTimeSelectorEnums, 64
- spinChunkGainSelectorEnums, 64
- spinChunkImageComponentEnums, 64
- spinChunkPixelFormatEnums, 65
- spinChunkRegionIDEnums, 65
- spinChunkScan3dCoordinateReferenceSelectorEnums, 66
- spinChunkScan3dCoordinateSelectorEnums, 66
- spinChunkScan3dCoordinateSystemEnums, 66
- spinChunkScan3dCoordinateSystemReferenceEnums, 67
- spinChunkScan3dCoordinateTransformSelectorEnums, 67
- spinChunkScan3dDistanceUnitEnums, 67

- spinChunkScan3dOutputModeEnums, 68
- spinChunkSelectorEnums, 68
- spinChunkSourceIDEnums, 69
- spinChunkTimerSelectorEnums, 69
- spinChunkTransferStreamIDEnums, 70
- spinCIConfigurationEnums, 70
- spinCITimeSlotsCountEnums, 70
- spinColorTransformationSelectorEnums, 71
- spinColorTransformationValueSelectorEnums, 71
- spinCompressionSaturationPriorityEnums, 72
- spinCounterEventActivationEnums, 72
- spinCounterEventSourceEnums, 72
- spinCounterResetActivationEnums, 73
- spinCounterResetSourceEnums, 73
- spinCounterSelectorEnums, 74
- spinCounterStatusEnums, 74
- spinCounterTriggerActivationEnums, 75
- spinCounterTriggerSourceEnums, 75
- spinCxpConnectionTestModeEnums, 76
- spinCxpLinkConfigurationEnums, 76
- spinCxpLinkConfigurationPreferredEnums, 77
- spinCxpLinkConfigurationStatusEnums, 78
- spinCxpPoCxpStatusEnums, 79
- spinDecimationHorizontalModeEnums, 79
- spinDecimationSelectorEnums, 79
- spinDecimationVerticalModeEnums, 80
- spinDefectCorrectionModeEnums, 80
- spinDeinterlacingEnums, 80
- spinDeviceCharacterSetEnums, 81
- spinDeviceClockSelectorEnums, 81
- spinDeviceConnectionStatusEnums, 81
- spinDeviceIndicatorModeEnums, 82
- spinDeviceLinkHeartbeatModeEnums, 82
- spinDeviceLinkThroughputLimitModeEnums, 82
- spinDevicePowerSupplySelectorEnums, 82
- spinDeviceRegistersEndiannessEnums, 83
- spinDeviceScanTypeEnums, 83
- spinDeviceSerialPortBaudRateEnums, 83
- spinDeviceSerialPortSelectorEnums, 84
- spinDeviceStreamChannelEndiannessEnums, 84
- spinDeviceStreamChannelTypeEnums, 84
- spinDeviceTapGeometryEnums, 85
- spinDeviceTemperatureSelectorEnums, 86
- spinDeviceTLTypeEnums, 86
- spinDeviceTypeEnums, 87
- spinEncoderModeEnums, 87
- spinEncoderOutputModeEnums, 87
- spinEncoderResetActivationEnums, 88
- spinEncoderResetSourceEnums, 88
- spinEncoderSelectorEnums, 89
- spinEncoderSourceAEnums, 90
- spinEncoderSourceBEnums, 90
- spinEncoderStatusEnums, 90
- spinEventNotificationEnums, 91
- spinEventSelectorEnums, 91
- spinExposureActiveModeEnums, 91
- spinExposureAutoEnums, 91
- spinExposureModeEnums, 92
- spinExposureTimeModeEnums, 92
- spinExposureTimeSelectorEnums, 93
- spinFileOpenModeEnums, 93
- spinFileOperationSelectorEnums, 93
- spinFileOperationStatusEnums, 94
- spinFileSelectorEnums, 94
- spinGainAutoBalanceEnums, 94
- spinGainAutoEnums, 96
- spinGainSelectorEnums, 96
- spinGevCCPEnums, 96
- spinGevCurrentPhysicalLinkConfigurationEnums, 97
- spinGevGVCPEExtendedStatusCodesSelectorEnums, 97
- spinGevGVSPExtendedIDModeEnums, 97
- spinGevIEEE1588ClockAccuracyEnums, 98
- spinGevIEEE1588ModeEnums, 98
- spinGevIEEE1588StatusEnums, 98
- spinGevIPConfigurationStatusEnums, 99
- spinGevPhysicalLinkConfigurationEnums, 99
- spinGevSupportedOptionSelectorEnums, 99
- spinImageComponentSelectorEnums, 100
- spinImageCompressionJPEGFormatOptionEnums, 101
- spinImageCompressionModeEnums, 101
- spinImageCompressionRateOptionEnums, 102
- spinLineFormatEnums, 102
- spinLineInputFilterSelectorEnums, 102
- spinLineModeEnums, 103
- spinLineSelectorEnums, 103
- spinLineSourceEnums, 103
- spinLogicBlockLUTInputActivationEnums, 104
- spinLogicBlockLUTInputSelectorEnums, 104
- spinLogicBlockLUTInputSourceEnums, 105
- spinLogicBlockLUTSelectorEnums, 105
- spinLogicBlockSelectorEnums, 106
- spinLUTSelectorEnums, 106
- spinPixelColorFilterEnums, 106
- spinPixelFormatEnums, 107
- spinPixelFormatInfoSelectorEnums, 112
- spinPixelSizeEnums, 118
- spinRegionDestinationEnums, 119
- spinRegionModeEnums, 119
- spinRegionSelectorEnums, 119
- spinRgbTransformLightSourceEnums, 120
- spinScan3dCoordinateReferenceSelectorEnums, 120
- spinScan3dCoordinateSelectorEnums, 121
- spinScan3dCoordinateSystemEnums, 121
- spinScan3dCoordinateSystemReferenceEnums, 121
- spinScan3dCoordinateTransformSelectorEnums, 122
- spinScan3dDistanceUnitEnums, 122
- spinScan3dOutputModeEnums, 122
- spinSensorDigitizationTapsEnums, 124
- spinSensorShutterModeEnums, 125
- spinSensorTapsEnums, 125

spinSequencerConfigurationModeEnums, 125
spinSequencerConfigurationValidEnums, 126
spinSequencerModeEnums, 126
spinSequencerSetValidEnums, 126
spinSequencerTriggerActivationEnums, 127
spinSequencerTriggerSourceEnums, 127
spinSerialPortBaudRateEnums, 127
spinSerialPortParityEnums, 128
spinSerialPortSelectorEnums, 128
spinSerialPortSourceEnums, 129
spinSerialPortStopBitsEnums, 129
spinSoftwareSignalSelectorEnums, 129
spinSourceSelectorEnums, 130
spinTestPatternEnums, 130
spinTestPatternGeneratorSelectorEnums, 130
spinTimerSelectorEnums, 131
spinTimerStatusEnums, 131
spinTimerTriggerActivationEnums, 131
spinTimerTriggerSourceEnums, 132
spinTransferComponentSelectorEnums, 133
spinTransferControlModeEnums, 133
spinTransferOperationModeEnums, 134
spinTransferQueueModeEnums, 134
spinTransferSelectorEnums, 134
spinTransferStatusSelectorEnums, 135
spinTransferTriggerActivationEnums, 135
spinTransferTriggerModeEnums, 135
spinTransferTriggerSelectorEnums, 136
spinTransferTriggerSourceEnums, 136
spinTriggerActivationEnums, 137
spinTriggerModeEnums, 138
spinTriggerOverlapEnums, 138
spinTriggerSelectorEnums, 138
spinTriggerSourceEnums, 138
spinUserOutputSelectorEnums, 139
spinUserSetDefaultEnums, 139
spinUserSetSelectorEnums, 140
spinWhiteClipSelectorEnums, 140
TestPattern_Increment, 130
TestPattern_Off, 130
TestPattern_SensorTestPattern, 130
TestPatternGeneratorSelector_PipelineStart, 130
TestPatternGeneratorSelector_Sensor, 130
TimerSelector_Timer0, 131
TimerSelector_Timer1, 131
TimerSelector_Timer2, 131
TimerStatus_TimerActive, 131
TimerStatus_TimerCompleted, 131
TimerStatus_TimerIdle, 131
TimerStatus_TimerTriggerWait, 131
TimerTriggerActivation_AnyEdge, 131
TimerTriggerActivation_FallingEdge, 131
TimerTriggerActivation_LevelHigh, 131
TimerTriggerActivation_LevelLow, 131
TimerTriggerActivation_RisingEdge, 131
TimerTriggerSource_AcquisitionEnd, 132
TimerTriggerSource_AcquisitionStart, 132
TimerTriggerSource_AcquisitionTrigger, 132
TimerTriggerSource_Action0, 133
TimerTriggerSource_Action1, 133
TimerTriggerSource_Action2, 133
TimerTriggerSource_Counter0End, 132
TimerTriggerSource_Counter0Start, 132
TimerTriggerSource_Counter1End, 132
TimerTriggerSource_Counter1Start, 132
TimerTriggerSource_Counter2End, 132
TimerTriggerSource_Counter2Start, 132
TimerTriggerSource_Encoder0, 133
TimerTriggerSource_Encoder1, 133
TimerTriggerSource_Encoder2, 133
TimerTriggerSource_ExposureEnd, 132
TimerTriggerSource_ExposureStart, 132
TimerTriggerSource_FrameBurstEnd, 132
TimerTriggerSource_FrameBurstStart, 132
TimerTriggerSource_FrameEnd, 132
TimerTriggerSource_FrameStart, 132
TimerTriggerSource_FrameTrigger, 132
TimerTriggerSource_Line0, 132
TimerTriggerSource_Line1, 132
TimerTriggerSource_Line2, 132
TimerTriggerSource_LineEnd, 132
TimerTriggerSource_LineStart, 132
TimerTriggerSource_LineTrigger, 132
TimerTriggerSource_LinkTrigger0, 133
TimerTriggerSource_LinkTrigger1, 133
TimerTriggerSource_LinkTrigger2, 133
TimerTriggerSource_Off, 132
TimerTriggerSource_SoftwareSignal0, 133
TimerTriggerSource_SoftwareSignal1, 133
TimerTriggerSource_SoftwareSignal2, 133
TimerTriggerSource_Timer0End, 132
TimerTriggerSource_Timer0Start, 132
TimerTriggerSource_Timer1End, 132
TimerTriggerSource_Timer1Start, 132
TimerTriggerSource_Timer2End, 132
TimerTriggerSource_Timer2Start, 132
TimerTriggerSource_UserOutput0, 132
TimerTriggerSource_UserOutput1, 132
TimerTriggerSource_UserOutput2, 132
TransferComponentSelector_All, 133
TransferComponentSelector_Blue, 133
TransferComponentSelector_Green, 133
TransferComponentSelector_Red, 133
TransferControlMode_Automatic, 134
TransferControlMode_Basic, 134
TransferControlMode_UserControlled, 134
TransferOperationMode_Continuous, 134
TransferOperationMode_MultiBlock, 134
TransferQueueMode_FirstInFirstOut, 134
TransferSelector_All, 134
TransferSelector_Stream0, 134
TransferSelector_Stream1, 134
TransferSelector_Stream2, 134
TransferStatusSelector_Paused, 135
TransferStatusSelector_QueueOverflow, 135
TransferStatusSelector_Stopped, 135

- TransferStatusSelector_Stopping, 135
- TransferStatusSelector_Streaming, 135
- TransferTriggerActivation_AnyEdge, 135
- TransferTriggerActivation_FallingEdge, 135
- TransferTriggerActivation_LevelHigh, 135
- TransferTriggerActivation_LevelLow, 135
- TransferTriggerActivation_RisingEdge, 135
- TransferTriggerMode_Off, 136
- TransferTriggerMode_On, 136
- TransferTriggerSelector_TransferAbort, 136
- TransferTriggerSelector_TransferActive, 136
- TransferTriggerSelector_TransferBurstStart, 136
- TransferTriggerSelector_TransferBurstStop, 136
- TransferTriggerSelector_TransferPause, 136
- TransferTriggerSelector_TransferResume, 136
- TransferTriggerSelector_TransferStart, 136
- TransferTriggerSelector_TransferStop, 136
- TransferTriggerSource_Action0, 137
- TransferTriggerSource_Action1, 137
- TransferTriggerSource_Action2, 137
- TransferTriggerSource_Counter0End, 137
- TransferTriggerSource_Counter0Start, 136
- TransferTriggerSource_Counter1End, 137
- TransferTriggerSource_Counter1Start, 136
- TransferTriggerSource_Counter2End, 137
- TransferTriggerSource_Counter2Start, 137
- TransferTriggerSource_Line0, 136
- TransferTriggerSource_Line1, 136
- TransferTriggerSource_Line2, 136
- TransferTriggerSource_SoftwareSignal0, 137
- TransferTriggerSource_SoftwareSignal1, 137
- TransferTriggerSource_SoftwareSignal2, 137
- TransferTriggerSource_Timer0End, 137
- TransferTriggerSource_Timer0Start, 137
- TransferTriggerSource_Timer1End, 137
- TransferTriggerSource_Timer1Start, 137
- TransferTriggerSource_Timer2End, 137
- TransferTriggerSource_Timer2Start, 137
- TriggerActivation_AnyEdge, 137
- TriggerActivation_FallingEdge, 137
- TriggerActivation_LevelHigh, 137
- TriggerActivation_LevelLow, 137
- TriggerActivation_RisingEdge, 137
- TriggerMode_Off, 138
- TriggerMode_On, 138
- TriggerOverlap_Off, 138
- TriggerOverlap_PreviousFrame, 138
- TriggerOverlap_ReadOut, 138
- TriggerSelector_AcquisitionStart, 138
- TriggerSelector_FrameBurstStart, 138
- TriggerSelector_FrameStart, 138
- TriggerSource_Action0, 139
- TriggerSource_Counter0End, 139
- TriggerSource_Counter0Start, 139
- TriggerSource_Counter1End, 139
- TriggerSource_Counter1Start, 139
- TriggerSource_Line0, 139
- TriggerSource_Line1, 139
- TriggerSource_Line2, 139
- TriggerSource_Line3, 139
- TriggerSource_LogicBlock0, 139
- TriggerSource_LogicBlock1, 139
- TriggerSource_Software, 139
- TriggerSource_UserOutput0, 139
- TriggerSource_UserOutput1, 139
- TriggerSource_UserOutput2, 139
- TriggerSource_UserOutput3, 139
- UNKNOWN_PIXELFORMAT, 112
- UserOutputSelector_UserOutput0, 139
- UserOutputSelector_UserOutput1, 139
- UserOutputSelector_UserOutput2, 139
- UserOutputSelector_UserOutput3, 139
- UserSetDefault_Default, 140
- UserSetDefault_UserSet0, 140
- UserSetDefault_UserSet1, 140
- UserSetSelector_Default, 140
- UserSetSelector_UserSet0, 140
- UserSetSelector_UserSet1, 140
- WhiteClipSelector_All, 140
- WhiteClipSelector_Blue, 140
- WhiteClipSelector_Green, 140
- WhiteClipSelector_Red, 140
- WhiteClipSelector_Tap1, 140
- WhiteClipSelector_Tap2, 140
- WhiteClipSelector_U, 140
- WhiteClipSelector_V, 140
- WhiteClipSelector_Y, 140
- CameraList Access, 143
- CategoryNode
 - SpinnakerGenApiDefsC.h, 511
- Chunk data access, 147
- Chunk Data Structures, 141
- ChunkBlackLevel
 - quickSpin, 181
- ChunkBlackLevelSelector
 - quickSpin, 182
- ChunkBlackLevelSelector_All
 - Camera Enumerations, 63
- ChunkCompressionMode
 - quickSpin, 182
- ChunkCompressionRatio
 - quickSpin, 182
- ChunkCounterSelector
 - quickSpin, 182
- ChunkCounterSelector_Counter0
 - Camera Enumerations, 63
- ChunkCounterSelector_Counter1
 - Camera Enumerations, 63
- ChunkCounterSelector_Counter2
 - Camera Enumerations, 63
- ChunkCounterValue
 - quickSpin, 182
- ChunkCRC
 - quickSpin, 182
- ChunkEnable
 - quickSpin, 182

- ChunkEncoderSelector
 - quickSpin, [182](#)
- ChunkEncoderSelector_Encoder0
 - Camera Enumerations, [63](#)
- ChunkEncoderSelector_Encoder1
 - Camera Enumerations, [63](#)
- ChunkEncoderSelector_Encoder2
 - Camera Enumerations, [63](#)
- ChunkEncoderStatus
 - quickSpin, [183](#)
- ChunkEncoderStatus_EncoderDown
 - Camera Enumerations, [63](#)
- ChunkEncoderStatus_EncoderIdle
 - Camera Enumerations, [63](#)
- ChunkEncoderStatus_EncoderStatic
 - Camera Enumerations, [63](#)
- ChunkEncoderStatus_EncoderUp
 - Camera Enumerations, [63](#)
- ChunkEncoderValue
 - quickSpin, [183](#)
- ChunkExposureEndLineStatusAll
 - quickSpin, [183](#)
- ChunkExposureTime
 - quickSpin, [183](#)
- ChunkExposureTimeSelector
 - quickSpin, [183](#)
- ChunkExposureTimeSelector_Blue
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Common
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Cyan
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Green
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Infrared
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Magenta
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Red
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Stage1
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Stage2
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Ultraviolet
 - Camera Enumerations, [64](#)
- ChunkExposureTimeSelector_Yellow
 - Camera Enumerations, [64](#)
- ChunkFrameID
 - quickSpin, [183](#)
- ChunkGain
 - quickSpin, [183](#)
- ChunkGainSelector
 - quickSpin, [183](#)
- ChunkGainSelector_All
 - Camera Enumerations, [64](#)
- ChunkGainSelector_Blue
 - Camera Enumerations, [64](#)
- ChunkGainSelector_Green
 - Camera Enumerations, [64](#)
- ChunkGainSelector_Red
 - Camera Enumerations, [64](#)
- ChunkHeight
 - quickSpin, [184](#)
- ChunkImage
 - quickSpin, [184](#)
- ChunkImageComponent
 - quickSpin, [184](#)
- ChunkImageComponent_Color
 - Camera Enumerations, [65](#)
- ChunkImageComponent_Confidence
 - Camera Enumerations, [65](#)
- ChunkImageComponent_Disparity
 - Camera Enumerations, [65](#)
- ChunkImageComponent_Infrared
 - Camera Enumerations, [65](#)
- ChunkImageComponent_Intensity
 - Camera Enumerations, [65](#)
- ChunkImageComponent_Range
 - Camera Enumerations, [65](#)
- ChunkImageComponent_Scatter
 - Camera Enumerations, [65](#)
- ChunkImageComponent_Ultraviolet
 - Camera Enumerations, [65](#)
- ChunkInferenceBoundingBoxResult
 - quickSpin, [184](#)
- ChunkInferenceConfidence
 - quickSpin, [184](#)
- ChunkInferenceFrameID
 - quickSpin, [184](#)
- ChunkInferenceResult
 - quickSpin, [184](#)
- ChunkLinePitch
 - quickSpin, [184](#)
- ChunkLineStatusAll
 - quickSpin, [185](#)
- ChunkModeActive
 - quickSpin, [185](#)
- ChunkOffsetX
 - quickSpin, [185](#)
- ChunkOffsetY
 - quickSpin, [185](#)
- ChunkPartSelector
 - quickSpin, [185](#)
- ChunkPixelDynamicRangeMax
 - quickSpin, [185](#)
- ChunkPixelDynamicRangeMin
 - quickSpin, [185](#)
- ChunkPixelFormat
 - quickSpin, [185](#)
- ChunkPixelFormat_BayerBG8
 - Camera Enumerations, [65](#)
- ChunkPixelFormat_BayerGB8
 - Camera Enumerations, [65](#)
- ChunkPixelFormat_BayerGR8
 - Camera Enumerations, [65](#)

- ChunkPixelFormat_BayerRG8
 - Camera Enumerations, [65](#)
- ChunkPixelFormat_Mono12Packed
 - Camera Enumerations, [65](#)
- ChunkPixelFormat_Mono16
 - Camera Enumerations, [65](#)
- ChunkPixelFormat_Mono8
 - Camera Enumerations, [65](#)
- ChunkPixelFormat_RGB8Packed
 - Camera Enumerations, [65](#)
- ChunkPixelFormat_YCbCr601_422_8_CbYCrY
 - Camera Enumerations, [65](#)
- ChunkPixelFormat_YUV422Packed
 - Camera Enumerations, [65](#)
- ChunkRegionID
 - quickSpin, [186](#)
- ChunkRegionID_Region0
 - Camera Enumerations, [65](#)
- ChunkRegionID_Region1
 - Camera Enumerations, [65](#)
- ChunkRegionID_Region2
 - Camera Enumerations, [65](#)
- ChunkScan3dAxisMax
 - quickSpin, [186](#)
- ChunkScan3dAxisMin
 - quickSpin, [186](#)
- ChunkScan3dCoordinateOffset
 - quickSpin, [186](#)
- ChunkScan3dCoordinateReferenceSelector
 - quickSpin, [186](#)
- ChunkScan3dCoordinateReferenceSelector_RotationX
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateReferenceSelector_RotationY
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateReferenceSelector_RotationZ
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationX
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationY
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationZ
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateReferenceValue
 - quickSpin, [186](#)
- ChunkScan3dCoordinateScale
 - quickSpin, [186](#)
- ChunkScan3dCoordinateSelector
 - quickSpin, [186](#)
- ChunkScan3dCoordinateSelector_CoordinateA
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateSelector_CoordinateB
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateSelector_CoordinateC
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateSystem
 - quickSpin, [187](#)
- ChunkScan3dCoordinateSystem_Cartesian
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateSystem_Cylindrical
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateSystem_Spherical
 - Camera Enumerations, [66](#)
- ChunkScan3dCoordinateSystemReference
 - quickSpin, [187](#)
- ChunkScan3dCoordinateSystemReference_Anchor
 - Camera Enumerations, [67](#)
- ChunkScan3dCoordinateSystemReference_Transformed
 - Camera Enumerations, [67](#)
- ChunkScan3dCoordinateTransformSelector
 - quickSpin, [187](#)
- ChunkScan3dCoordinateTransformSelector_RotationX
 - Camera Enumerations, [67](#)
- ChunkScan3dCoordinateTransformSelector_RotationY
 - Camera Enumerations, [67](#)
- ChunkScan3dCoordinateTransformSelector_RotationZ
 - Camera Enumerations, [67](#)
- ChunkScan3dCoordinateTransformSelector_TranslationX
 - Camera Enumerations, [67](#)
- ChunkScan3dCoordinateTransformSelector_TranslationY
 - Camera Enumerations, [67](#)
- ChunkScan3dCoordinateTransformSelector_TranslationZ
 - Camera Enumerations, [67](#)
- ChunkScan3dDistanceUnit
 - quickSpin, [187](#)
- ChunkScan3dDistanceUnit_Inch
 - Camera Enumerations, [67](#)
- ChunkScan3dDistanceUnit_Millimeter
 - Camera Enumerations, [67](#)
- ChunkScan3dInvalidDataFlag
 - quickSpin, [187](#)
- ChunkScan3dInvalidDataValue
 - quickSpin, [187](#)
- ChunkScan3dOutputMode
 - quickSpin, [187](#)
- ChunkScan3dOutputMode_CalibratedABC_Grid
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_CalibratedABC_PointCloud
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_CalibratedAC
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_CalibratedAC_Linescan
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_CalibratedC
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_CalibratedC_Linescan
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_DisparityC
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_DisparityC_Linescan
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_RectifiedC
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_RectifiedC_Linescan
 - Camera Enumerations, [68](#)
- ChunkScan3dOutputMode_UncalibratedC
 - Camera Enumerations, [68](#)

- ChunkScan3dTransformValue
 - quickSpin, [187](#)
- ChunkScanLineSelector
 - quickSpin, [188](#)
- ChunkSelector
 - quickSpin, [188](#)
- ChunkSelector_BlackLevel
 - Camera Enumerations, [69](#)
- ChunkSelector_CRC
 - Camera Enumerations, [69](#)
- ChunkSelector_ExposureEndLineStatusAll
 - Camera Enumerations, [69](#)
- ChunkSelector_ExposureTime
 - Camera Enumerations, [69](#)
- ChunkSelector_FrameID
 - Camera Enumerations, [69](#)
- ChunkSelector_Gain
 - Camera Enumerations, [69](#)
- ChunkSelector_Height
 - Camera Enumerations, [69](#)
- ChunkSelector_Image
 - Camera Enumerations, [69](#)
- ChunkSelector_OffsetX
 - Camera Enumerations, [69](#)
- ChunkSelector_OffsetY
 - Camera Enumerations, [69](#)
- ChunkSelector_PixelFormat
 - Camera Enumerations, [69](#)
- ChunkSelector_SequencerSetActive
 - Camera Enumerations, [69](#)
- ChunkSelector_SerialData
 - Camera Enumerations, [69](#)
- ChunkSelector_Timestamp
 - Camera Enumerations, [69](#)
- ChunkSelector_Width
 - Camera Enumerations, [69](#)
- ChunkSequencerSetActive
 - quickSpin, [188](#)
- ChunkSerialData
 - quickSpin, [188](#)
- ChunkSerialDataLength
 - quickSpin, [188](#)
- ChunkSerialReceiveOverflow
 - quickSpin, [188](#)
- ChunkSourceID
 - quickSpin, [188](#)
- ChunkSourceID_Source0
 - Camera Enumerations, [69](#)
- ChunkSourceID_Source1
 - Camera Enumerations, [69](#)
- ChunkSourceID_Source2
 - Camera Enumerations, [69](#)
- ChunkStreamChannelID
 - quickSpin, [188](#)
- ChunkTimerSelector
 - quickSpin, [189](#)
- ChunkTimerSelector_Timer0
 - Camera Enumerations, [70](#)
- ChunkTimerSelector_Timer1
 - Camera Enumerations, [70](#)
- ChunkTimerSelector_Timer2
 - Camera Enumerations, [70](#)
- ChunkTimerValue
 - quickSpin, [189](#)
- ChunkTimestamp
 - quickSpin, [189](#)
- ChunkTimestampLatchValue
 - quickSpin, [189](#)
- ChunkTransferBlockID
 - quickSpin, [189](#)
- ChunkTransferQueueCurrentBlockCount
 - quickSpin, [189](#)
- ChunkTransferStreamID
 - quickSpin, [189](#)
- ChunkTransferStreamID_Stream0
 - Camera Enumerations, [70](#)
- ChunkTransferStreamID_Stream1
 - Camera Enumerations, [70](#)
- ChunkTransferStreamID_Stream2
 - Camera Enumerations, [70](#)
- ChunkTransferStreamID_Stream3
 - Camera Enumerations, [70](#)
- ChunkWidth
 - quickSpin, [189](#)
- CL
 - SpinnakerGenApiDefsC.h, [512](#)
- CIConfiguration
 - quickSpin, [190](#)
- CIConfiguration_Base
 - Camera Enumerations, [70](#)
- CIConfiguration_DualBase
 - Camera Enumerations, [70](#)
- CIConfiguration_EightyBit
 - Camera Enumerations, [70](#)
- CIConfiguration_Full
 - Camera Enumerations, [70](#)
- CIConfiguration_Medium
 - Camera Enumerations, [70](#)
- CITimeSlotsCount
 - quickSpin, [190](#)
- CITimeSlotsCount_One
 - Camera Enumerations, [71](#)
- CITimeSlotsCount_Three
 - Camera Enumerations, [71](#)
- CITimeSlotsCount_Two
 - Camera Enumerations, [71](#)
- ColorTransformationEnable
 - quickSpin, [190](#)
- ColorTransformationSelector
 - quickSpin, [190](#)
- ColorTransformationSelector_RGBtoRGB
 - Camera Enumerations, [71](#)
- ColorTransformationSelector_RGBtoYUV
 - Camera Enumerations, [71](#)
- ColorTransformationValue
 - quickSpin, [190](#)

- ColorTransformationValueSelector
 - quickSpin, [190](#)
- ColorTransformationValueSelector_Gain00
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Gain01
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Gain02
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Gain10
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Gain11
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Gain12
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Gain20
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Gain21
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Gain22
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Offset0
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Offset1
 - Camera Enumerations, [71](#)
- ColorTransformationValueSelector_Offset2
 - Camera Enumerations, [71](#)
- CommandNode
 - SpinnakerGenApiDefsC.h, [510](#)
- compression
 - spinTIFFOption, [301](#)
- compressionLevel
 - spinPNGOption, [299](#)
- CompressionRatio
 - quickSpin, [190](#)
- CompressionSaturationPriority
 - quickSpin, [190](#)
- CompressionSaturationPriority_DropFrame
 - Camera Enumerations, [72](#)
- CompressionSaturationPriority_ReduceFrameRate
 - Camera Enumerations, [72](#)
- CounterDelay
 - quickSpin, [191](#)
- CounterDuration
 - quickSpin, [191](#)
- CounterEventActivation
 - quickSpin, [191](#)
- CounterEventActivation_AnyEdge
 - Camera Enumerations, [72](#)
- CounterEventActivation_FallingEdge
 - Camera Enumerations, [72](#)
- CounterEventActivation_LevelHigh
 - Camera Enumerations, [72](#)
- CounterEventActivation_LevelLow
 - Camera Enumerations, [72](#)
- CounterEventActivation_RisingEdge
 - Camera Enumerations, [72](#)
- CounterEventSource
 - quickSpin, [191](#)
- CounterEventSource_Counter0End
 - Camera Enumerations, [73](#)
- CounterEventSource_Counter0Start
 - Camera Enumerations, [73](#)
- CounterEventSource_Counter1End
 - Camera Enumerations, [73](#)
- CounterEventSource_Counter1Start
 - Camera Enumerations, [73](#)
- CounterEventSource_ExposureEnd
 - Camera Enumerations, [73](#)
- CounterEventSource_ExposureStart
 - Camera Enumerations, [73](#)
- CounterEventSource_FrameTriggerWait
 - Camera Enumerations, [73](#)
- CounterEventSource_Line0
 - Camera Enumerations, [72](#)
- CounterEventSource_Line1
 - Camera Enumerations, [73](#)
- CounterEventSource_Line2
 - Camera Enumerations, [73](#)
- CounterEventSource_Line3
 - Camera Enumerations, [73](#)
- CounterEventSource_LogicBlock0
 - Camera Enumerations, [73](#)
- CounterEventSource_LogicBlock1
 - Camera Enumerations, [73](#)
- CounterEventSource_MHzTick
 - Camera Enumerations, [72](#)
- CounterEventSource_Off
 - Camera Enumerations, [72](#)
- CounterEventSource_UserOutput0
 - Camera Enumerations, [73](#)
- CounterEventSource_UserOutput1
 - Camera Enumerations, [73](#)
- CounterEventSource_UserOutput2
 - Camera Enumerations, [73](#)
- CounterEventSource_UserOutput3
 - Camera Enumerations, [73](#)
- CounterReset
 - quickSpin, [191](#)
- CounterResetActivation
 - quickSpin, [191](#)
- CounterResetActivation_AnyEdge
 - Camera Enumerations, [73](#)
- CounterResetActivation_FallingEdge
 - Camera Enumerations, [73](#)
- CounterResetActivation_LevelHigh
 - Camera Enumerations, [73](#)
- CounterResetActivation_LevelLow
 - Camera Enumerations, [73](#)
- CounterResetActivation_RisingEdge
 - Camera Enumerations, [73](#)
- CounterResetSource
 - quickSpin, [191](#)
- CounterResetSource_Counter0End
 - Camera Enumerations, [74](#)
- CounterResetSource_Counter0Start
 - Camera Enumerations, [74](#)

- CounterResetSource_Counter1End
 - Camera Enumerations, [74](#)
- CounterResetSource_Counter1Start
 - Camera Enumerations, [74](#)
- CounterResetSource_ExposureEnd
 - Camera Enumerations, [74](#)
- CounterResetSource_ExposureStart
 - Camera Enumerations, [74](#)
- CounterResetSource_FrameTriggerWait
 - Camera Enumerations, [74](#)
- CounterResetSource_Line0
 - Camera Enumerations, [74](#)
- CounterResetSource_Line1
 - Camera Enumerations, [74](#)
- CounterResetSource_Line2
 - Camera Enumerations, [74](#)
- CounterResetSource_Line3
 - Camera Enumerations, [74](#)
- CounterResetSource_LogicBlock0
 - Camera Enumerations, [74](#)
- CounterResetSource_LogicBlock1
 - Camera Enumerations, [74](#)
- CounterResetSource_Off
 - Camera Enumerations, [73](#)
- CounterResetSource_UserOutput0
 - Camera Enumerations, [74](#)
- CounterResetSource_UserOutput1
 - Camera Enumerations, [74](#)
- CounterResetSource_UserOutput2
 - Camera Enumerations, [74](#)
- CounterResetSource_UserOutput3
 - Camera Enumerations, [74](#)
- CounterSelector
 - quickSpin, [191](#)
- CounterSelector_Counter0
 - Camera Enumerations, [74](#)
- CounterSelector_Counter1
 - Camera Enumerations, [74](#)
- CounterStatus
 - quickSpin, [192](#)
- CounterStatus_CounterActive
 - Camera Enumerations, [74](#)
- CounterStatus_CounterCompleted
 - Camera Enumerations, [74](#)
- CounterStatus_CounterIdle
 - Camera Enumerations, [74](#)
- CounterStatus_CounterOverflow
 - Camera Enumerations, [74](#)
- CounterStatus_CounterTriggerWait
 - Camera Enumerations, [74](#)
- CounterTriggerActivation
 - quickSpin, [192](#)
- CounterTriggerActivation_AnyEdge
 - Camera Enumerations, [75](#)
- CounterTriggerActivation_FallingEdge
 - Camera Enumerations, [75](#)
- CounterTriggerActivation_LevelHigh
 - Camera Enumerations, [75](#)
- CounterTriggerActivation_LevelLow
 - Camera Enumerations, [75](#)
- CounterTriggerActivation_RisingEdge
 - Camera Enumerations, [75](#)
- CounterTriggerSource
 - quickSpin, [192](#)
- CounterTriggerSource_Counter0End
 - Camera Enumerations, [75](#)
- CounterTriggerSource_Counter0Start
 - Camera Enumerations, [75](#)
- CounterTriggerSource_Counter1End
 - Camera Enumerations, [75](#)
- CounterTriggerSource_Counter1Start
 - Camera Enumerations, [75](#)
- CounterTriggerSource_ExposureEnd
 - Camera Enumerations, [75](#)
- CounterTriggerSource_ExposureStart
 - Camera Enumerations, [75](#)
- CounterTriggerSource_FrameTriggerWait
 - Camera Enumerations, [75](#)
- CounterTriggerSource_Line0
 - Camera Enumerations, [75](#)
- CounterTriggerSource_Line1
 - Camera Enumerations, [75](#)
- CounterTriggerSource_Line2
 - Camera Enumerations, [75](#)
- CounterTriggerSource_Line3
 - Camera Enumerations, [75](#)
- CounterTriggerSource_LogicBlock0
 - Camera Enumerations, [75](#)
- CounterTriggerSource_LogicBlock1
 - Camera Enumerations, [75](#)
- CounterTriggerSource_Off
 - Camera Enumerations, [75](#)
- CounterTriggerSource_UserOutput0
 - Camera Enumerations, [75](#)
- CounterTriggerSource_UserOutput1
 - Camera Enumerations, [75](#)
- CounterTriggerSource_UserOutput2
 - Camera Enumerations, [75](#)
- CounterTriggerSource_UserOutput3
 - Camera Enumerations, [75](#)
- CounterValue
 - quickSpin, [192](#)
- CounterValueAtReset
 - quickSpin, [192](#)
- ctAllDependingNodes
 - SpinnakerGenApiDefsC.h, [510](#)
- ctAllTerminalNodes
 - SpinnakerGenApiDefsC.h, [510](#)
- ctDependingChildren
 - SpinnakerGenApiDefsC.h, [510](#)
- ctInvalidators
 - SpinnakerGenApiDefsC.h, [510](#)
- ctReadingChildren
 - SpinnakerGenApiDefsC.h, [510](#)
- ctWritingChildren
 - SpinnakerGenApiDefsC.h, [510](#)

- Custom
 - SpinnakerGenApiDefsC.h, [510](#)
- CxpConnectionSelector
 - quickSpin, [192](#)
- CxpConnectionTestErrorCount
 - quickSpin, [192](#)
- CxpConnectionTestMode
 - quickSpin, [192](#)
- CxpConnectionTestMode_Mode1
 - Camera Enumerations, [76](#)
- CxpConnectionTestMode_Off
 - Camera Enumerations, [76](#)
- CxpConnectionTestPacketCount
 - quickSpin, [193](#)
- CxpLinkConfiguration
 - quickSpin, [193](#)
- CxpLinkConfiguration_Auto
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP1_X1
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP1_X2
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP1_X3
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP1_X4
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP1_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP1_X6
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP2_X1
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP2_X2
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP2_X3
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP2_X4
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP2_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP2_X6
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP3_X1
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP3_X2
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP3_X3
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP3_X4
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP3_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP3_X6
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP5_X1
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP5_X2
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP5_X3
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP5_X4
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP5_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP5_X6
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP6_X1
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP6_X2
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP6_X3
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP6_X4
 - Camera Enumerations, [76](#)
- CxpLinkConfiguration_CXP6_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfiguration_CXP6_X6
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred
 - quickSpin, [193](#)
- CxpLinkConfigurationPreferred_CXP1_X1
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP1_X2
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP1_X3
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP1_X4
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP1_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP1_X6
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationPreferred_CXP2_X1
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP2_X2
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP2_X3
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP2_X4
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP2_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP2_X6
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationPreferred_CXP3_X1
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP3_X2
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP3_X3
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP3_X4
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP3_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP3_X6
 - Camera Enumerations, [78](#)

- CxpLinkConfigurationPreferred_CXP5_X1
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP5_X2
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP5_X3
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP5_X4
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP5_X5
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP5_X6
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationPreferred_CXP6_X1
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP6_X2
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP6_X3
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP6_X4
 - Camera Enumerations, [77](#)
- CxpLinkConfigurationPreferred_CXP6_X5
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationPreferred_CXP6_X6
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus
 - quickSpin, [193](#)
- CxpLinkConfigurationStatus_CXP1_X1
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP1_X2
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP1_X3
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP1_X4
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP1_X5
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP1_X6
 - Camera Enumerations, [79](#)
- CxpLinkConfigurationStatus_CXP2_X1
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP2_X2
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP2_X3
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP2_X4
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP2_X5
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP2_X6
 - Camera Enumerations, [79](#)
- CxpLinkConfigurationStatus_CXP3_X1
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP3_X2
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP3_X3
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP3_X4
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP3_X5
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP3_X6
 - Camera Enumerations, [79](#)
- CxpLinkConfigurationStatus_CXP5_X1
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP5_X2
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP5_X3
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP5_X4
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP5_X5
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP5_X6
 - Camera Enumerations, [79](#)
- CxpLinkConfigurationStatus_CXP6_X1
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP6_X2
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP6_X3
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP6_X4
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_CXP6_X5
 - Camera Enumerations, [79](#)
- CxpLinkConfigurationStatus_CXP6_X6
 - Camera Enumerations, [79](#)
- CxpLinkConfigurationStatus_None
 - Camera Enumerations, [78](#)
- CxpLinkConfigurationStatus_Pending
 - Camera Enumerations, [78](#)
- CxpPoCxpAuto
 - quickSpin, [193](#)
- CxpPoCxpStatus
 - quickSpin, [193](#)
- CxpPoCxpStatus_Auto
 - Camera Enumerations, [79](#)
- CxpPoCxpStatus_Off
 - Camera Enumerations, [79](#)
- CxpPoCxpStatus_Tripped
 - Camera Enumerations, [79](#)
- CxpPoCxpTripReset
 - quickSpin, [193](#)
- CxpPoCxpTurnOff
 - quickSpin, [193](#)
- DecimationHorizontal
 - quickSpin, [194](#)
- DecimationHorizontalMode
 - quickSpin, [194](#)
- DecimationHorizontalMode_Discard
 - Camera Enumerations, [79](#)
- DecimationSelector
 - quickSpin, [194](#)
- DecimationSelector_All
 - Camera Enumerations, [80](#)
- DecimationSelector_Sensor
 - Camera Enumerations, [80](#)

- DecimationVertical
 - quickSpin, [194](#)
- DecimationVerticalMode
 - quickSpin, [194](#)
- DecimationVerticalMode_Discard
 - Camera Enumerations, [80](#)
- Decreasing
 - SpinnakerGenApiDefsC.h, [512](#)
- DefectCorrectionMode
 - quickSpin, [194](#)
- DefectCorrectionMode_Average
 - Camera Enumerations, [80](#)
- DefectCorrectionMode_Highlight
 - Camera Enumerations, [80](#)
- DefectCorrectionMode_Zero
 - Camera Enumerations, [80](#)
- DefectCorrectStaticEnable
 - quickSpin, [194](#)
- DefectTableApply
 - quickSpin, [194](#)
- DefectTableCoordinateX
 - quickSpin, [195](#)
- DefectTableCoordinateY
 - quickSpin, [195](#)
- DefectTableFactoryRestore
 - quickSpin, [195](#)
- DefectTableIndex
 - quickSpin, [195](#)
- DefectTablePixelCount
 - quickSpin, [195](#)
- DefectTableSave
 - quickSpin, [195](#)
- Deinterlacing
 - quickSpin, [195](#)
- Deinterlacing_LineDuplication
 - Camera Enumerations, [81](#)
- Deinterlacing_Off
 - Camera Enumerations, [81](#)
- Deinterlacing_Weave
 - Camera Enumerations, [81](#)
- Device Event Data Access, [147](#)
- DeviceAccessStatus
 - quickSpinTLDevice, [259](#)
 - quickSpinTLInterface, [266](#)
- DeviceAccessStatus_Busy
 - Transport Layer Enumerations, [153](#)
- DeviceAccessStatus_NoAccess
 - Transport Layer Enumerations, [153](#)
- DeviceAccessStatus_OpenReadOnly
 - Transport Layer Enumerations, [153](#)
- DeviceAccessStatus_OpenReadWrite
 - Transport Layer Enumerations, [153](#)
- DeviceAccessStatus_ReadOnly
 - Transport Layer Enumerations, [153](#)
- DeviceAccessStatus_ReadWrite
 - Transport Layer Enumerations, [153](#)
- DeviceAccessStatus_Unknown
 - Transport Layer Enumerations, [153](#)
- DeviceAddress
 - actionCommandResult, [161](#)
- DeviceBootloaderVersion
 - quickSpinTLDevice, [260](#)
- DeviceCharacterSet
 - quickSpin, [195](#)
- DeviceCharacterSet_ASCII
 - Camera Enumerations, [81](#)
- DeviceCharacterSet_UTF8
 - Camera Enumerations, [81](#)
- DeviceClockFrequency
 - quickSpin, [196](#)
- DeviceClockSelector
 - quickSpin, [196](#)
- DeviceClockSelector_CameraLink
 - Camera Enumerations, [81](#)
- DeviceClockSelector_Sensor
 - Camera Enumerations, [81](#)
- DeviceClockSelector_SensorDigitization
 - Camera Enumerations, [81](#)
- DeviceConnectionSelector
 - quickSpin, [196](#)
- DeviceConnectionSpeed
 - quickSpin, [196](#)
- DeviceConnectionStatus
 - quickSpin, [196](#)
- DeviceConnectionStatus_Active
 - Camera Enumerations, [81](#)
- DeviceConnectionStatus_Inactive
 - Camera Enumerations, [81](#)
- DeviceCount
 - quickSpinTLInterface, [266](#)
- DeviceCurrentSpeed
 - quickSpinTLDevice, [260](#)
- DeviceCurrentSpeed_FullSpeed
 - Transport Layer Enumerations, [154](#)
- DeviceCurrentSpeed_HighSpeed
 - Transport Layer Enumerations, [154](#)
- DeviceCurrentSpeed_LowSpeed
 - Transport Layer Enumerations, [154](#)
- DeviceCurrentSpeed_SuperSpeed
 - Transport Layer Enumerations, [154](#)
- DeviceCurrentSpeed_UnknownSpeed
 - Transport Layer Enumerations, [154](#)
- DeviceDisplayName
 - quickSpinTLDevice, [260](#)
- DeviceDriverVersion
 - quickSpinTLDevice, [260](#)
- DeviceEndiannessMechanism
 - quickSpinTLDevice, [260](#)
- DeviceEndiannessMechanism_Legacy
 - Transport Layer Enumerations, [154](#)
- DeviceEndiannessMechanism_Standard
 - Transport Layer Enumerations, [154](#)
- DeviceEventChannelCount
 - quickSpin, [196](#)
- DeviceFamilyName
 - quickSpin, [196](#)

- DeviceFeaturePersistenceEnd
 - quickSpin, [196](#)
- DeviceFeaturePersistenceStart
 - quickSpin, [197](#)
- DeviceFirmwareVersion
 - quickSpin, [197](#)
- DeviceGenCPVersionMajor
 - quickSpin, [197](#)
- DeviceGenCPVersionMinor
 - quickSpin, [197](#)
- DeviceID
 - quickSpin, [197](#)
 - quickSpinTLDevice, [260](#)
 - quickSpinTLInterface, [267](#)
- DeviceIndicatorMode
 - quickSpin, [197](#)
- DeviceIndicatorMode_Active
 - Camera Enumerations, [82](#)
- DeviceIndicatorMode_ErrorStatus
 - Camera Enumerations, [82](#)
- DeviceIndicatorMode_Inactive
 - Camera Enumerations, [82](#)
- DeviceInstanceId
 - quickSpinTLDevice, [260](#)
- DevicesUpdater
 - quickSpinTLDevice, [260](#)
- DeviceLinkBandwidthReserve
 - quickSpin, [197](#)
- DeviceLinkCommandTimeout
 - quickSpin, [197](#)
- DeviceLinkConnectionCount
 - quickSpin, [198](#)
- DeviceLinkCurrentThroughput
 - quickSpin, [198](#)
- DeviceLinkHeartbeatMode
 - quickSpin, [198](#)
- DeviceLinkHeartbeatMode_Off
 - Camera Enumerations, [82](#)
- DeviceLinkHeartbeatMode_On
 - Camera Enumerations, [82](#)
- DeviceLinkHeartbeatTimeout
 - quickSpin, [198](#)
- DeviceLinkSelector
 - quickSpin, [198](#)
- DeviceLinkSpeed
 - quickSpin, [198](#)
 - quickSpinTLDevice, [261](#)
- DeviceLinkThroughputLimit
 - quickSpin, [198](#)
- DeviceLinkThroughputLimitMode
 - quickSpin, [198](#)
- DeviceLinkThroughputLimitMode_Off
 - Camera Enumerations, [82](#)
- DeviceLinkThroughputLimitMode_On
 - Camera Enumerations, [82](#)
- DeviceLocation
 - quickSpinTLDevice, [261](#)
- DeviceManifestEntrySelector
 - quickSpin, [199](#)
- DeviceManifestPrimaryURL
 - quickSpin, [199](#)
- DeviceManifestSchemaMajorVersion
 - quickSpin, [199](#)
- DeviceManifestSchemaMinorVersion
 - quickSpin, [199](#)
- DeviceManifestSecondaryURL
 - quickSpin, [199](#)
- DeviceManifestXMLMajorVersion
 - quickSpin, [199](#)
- DeviceManifestXMLMinorVersion
 - quickSpin, [199](#)
- DeviceManifestXMLSubMinorVersion
 - quickSpin, [199](#)
- DeviceManufacturerInfo
 - quickSpin, [200](#)
- DeviceMaxThroughput
 - quickSpin, [200](#)
- DeviceModelName
 - quickSpin, [200](#)
 - quickSpinTLDevice, [261](#)
 - quickSpinTLInterface, [267](#)
- DeviceMulticastMonitorMode
 - quickSpinTLDevice, [261](#)
- DevicePortId
 - quickSpinTLDevice, [261](#)
- DevicePowerSupplySelector
 - quickSpin, [200](#)
- DevicePowerSupplySelector_External
 - Camera Enumerations, [83](#)
- DeviceRegistersCheck
 - quickSpin, [200](#)
- DeviceRegistersEndianness
 - quickSpin, [200](#)
- DeviceRegistersEndianness_Big
 - Camera Enumerations, [83](#)
- DeviceRegistersEndianness_Little
 - Camera Enumerations, [83](#)
- DeviceRegistersStreamingEnd
 - quickSpin, [200](#)
- DeviceRegistersStreamingStart
 - quickSpin, [200](#)
- DeviceRegistersValid
 - quickSpin, [201](#)
- DeviceReset
 - quickSpin, [201](#)
 - quickSpinTLDevice, [261](#)
- DeviceScanType
 - quickSpin, [201](#)
- DeviceScanType_Areascan
 - Camera Enumerations, [83](#)
- DeviceSelector
 - quickSpinTLInterface, [267](#)
- DeviceSerialNumber
 - quickSpin, [201](#)
 - quickSpinTLDevice, [261](#)
 - quickSpinTLInterface, [267](#)

- DeviceSerialPortBaudRate
 - quickSpin, [201](#)
- DeviceSerialPortBaudRate_Baud115200
 - Camera Enumerations, [84](#)
- DeviceSerialPortBaudRate_Baud19200
 - Camera Enumerations, [84](#)
- DeviceSerialPortBaudRate_Baud230400
 - Camera Enumerations, [84](#)
- DeviceSerialPortBaudRate_Baud38400
 - Camera Enumerations, [84](#)
- DeviceSerialPortBaudRate_Baud460800
 - Camera Enumerations, [84](#)
- DeviceSerialPortBaudRate_Baud57600
 - Camera Enumerations, [84](#)
- DeviceSerialPortBaudRate_Baud921600
 - Camera Enumerations, [84](#)
- DeviceSerialPortBaudRate_Baud9600
 - Camera Enumerations, [84](#)
- DeviceSerialPortSelector
 - quickSpin, [201](#)
- DeviceSerialPortSelector_CameraLink
 - Camera Enumerations, [84](#)
- DeviceSFNCVersionMajor
 - quickSpin, [201](#)
- DeviceSFNCVersionMinor
 - quickSpin, [201](#)
- DeviceSFNCVersionSubMinor
 - quickSpin, [202](#)
- DeviceStreamChannelCount
 - quickSpin, [202](#)
- DeviceStreamChannelEndianness
 - quickSpin, [202](#)
- DeviceStreamChannelEndianness_Big
 - Camera Enumerations, [84](#)
- DeviceStreamChannelEndianness_Little
 - Camera Enumerations, [84](#)
- DeviceStreamChannelLink
 - quickSpin, [202](#)
- DeviceStreamChannelPacketSize
 - quickSpin, [202](#)
- DeviceStreamChannelSelector
 - quickSpin, [202](#)
- DeviceStreamChannelType
 - quickSpin, [202](#)
- DeviceStreamChannelType_Receiver
 - Camera Enumerations, [85](#)
- DeviceStreamChannelType_Transmitter
 - Camera Enumerations, [85](#)
- DeviceTapGeometry
 - quickSpin, [202](#)
- DeviceTapGeometry_Geometry_10X
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_10X_1Y
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_1X
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X10
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_1X10_1Y
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_1X2
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X2_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X2_1Y2
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X2_2YE
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X3
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X3_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X4
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X4_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X8
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_1X8_1Y
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_1X_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X_1Y2
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_1X_2YE
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2X
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2X2
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2X2_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2X2E
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2X2E_2YE
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_2X2M
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_2X_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2X_2YE
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2XE
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2XE_1Y2
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2XE_2YE
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2XM
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2XM_1Y
 - Camera Enumerations, [85](#)

- DeviceTapGeometry_Geometry_2XM_1Y2
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_2XM_2YE
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_3X
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_3X_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_4X
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_4X2
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_4X2_1Y
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_4X2E
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_4X2E_1Y
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_4X_1Y
 - Camera Enumerations, [85](#)
- DeviceTapGeometry_Geometry_8X
 - Camera Enumerations, [86](#)
- DeviceTapGeometry_Geometry_8X_1Y
 - Camera Enumerations, [86](#)
- DeviceTemperature
 - quickSpin, [203](#)
- DeviceTemperatureSelector
 - quickSpin, [203](#)
- DeviceTemperatureSelector_Sensor
 - Camera Enumerations, [86](#)
- DeviceTLType
 - quickSpin, [203](#)
- DeviceTLType_CameraLink
 - Camera Enumerations, [86](#)
- DeviceTLType_CameraLinkHS
 - Camera Enumerations, [86](#)
- DeviceTLType_CoaXPress
 - Camera Enumerations, [86](#)
- DeviceTLType_Custom
 - Camera Enumerations, [86](#)
- DeviceTLType_GigEVision
 - Camera Enumerations, [86](#)
- DeviceTLType_USB3Vision
 - Camera Enumerations, [86](#)
- DeviceTLVersionMajor
 - quickSpin, [203](#)
- DeviceTLVersionMinor
 - quickSpin, [203](#)
- DeviceTLVersionSubMinor
 - quickSpin, [203](#)
- DeviceType
 - quickSpin, [203](#)
 - quickSpinTLDevice, [261](#)
- DeviceType_CameraLink
 - Transport Layer Enumerations, [154](#)
- DeviceType_CameraLinkHS
 - Transport Layer Enumerations, [154](#)
- DeviceType_CoaXPress
 - Transport Layer Enumerations, [154](#)
- DeviceType_Custom
 - Transport Layer Enumerations, [154](#)
- DeviceType_GigEVision
 - Transport Layer Enumerations, [154](#)
- DeviceType_Peripheral
 - Camera Enumerations, [87](#)
- DeviceType_Receiver
 - Camera Enumerations, [87](#)
- DeviceType_Transceiver
 - Camera Enumerations, [87](#)
- DeviceType_Transmitter
 - Camera Enumerations, [87](#)
- DeviceType_USB3Vision
 - Transport Layer Enumerations, [154](#)
- DeviceU3VProtocol
 - quickSpinTLDevice, [262](#)
- DeviceUnlock
 - quickSpinTLInterface, [267](#)
- DeviceUpdateList
 - quickSpinTLInterface, [267](#)
- DeviceUptime
 - quickSpin, [203](#)
- DeviceUserID
 - quickSpin, [204](#)
 - quickSpinTLDevice, [262](#)
- DeviceVendorName
 - quickSpin, [204](#)
 - quickSpinTLDevice, [262](#)
 - quickSpinTLInterface, [267](#)
- DeviceVersion
 - quickSpin, [204](#)
 - quickSpinTLDevice, [262](#)
- doc/spindocs/C/GettingStarted.dox, [303](#)
- doc/spindocs/C/ProgrammerGuide.dox, [303](#)
- doc/spindocs/shared/Benefits.dox, [303](#)
- doc/spindocs/shared/FlyCapture2Comparison.dox, [303](#)
- doc/spindocs/shared/GenICamGenTL.dox, [303](#)
- doc/spindocs/shared/Licensing.dox, [303](#)
- doc/spindocs/shared/Maintenance.dox, [303](#)
- doc/spindocs/shared/NetworkingBestPractices.dox, [303](#)
- EncoderDivider
 - quickSpin, [204](#)
- EncoderMode
 - quickSpin, [204](#)
- EncoderMode_FourPhase
 - Camera Enumerations, [87](#)
- EncoderMode_HighResolution
 - Camera Enumerations, [87](#)
- EncoderOutputMode
 - quickSpin, [204](#)
- EncoderOutputMode_DirectionDown
 - Camera Enumerations, [88](#)
- EncoderOutputMode_DirectionUp
 - Camera Enumerations, [88](#)
- EncoderOutputMode_Motion
 - Camera Enumerations, [88](#)
- EncoderOutputMode_Off

- Camera Enumerations, [87](#)
- EncoderOutputMode_PositionDown
 - Camera Enumerations, [88](#)
- EncoderOutputMode_PositionUp
 - Camera Enumerations, [87](#)
- EncoderReset
 - quickSpin, [204](#)
- EncoderResetActivation
 - quickSpin, [204](#)
- EncoderResetActivation_AnyEdge
 - Camera Enumerations, [88](#)
- EncoderResetActivation_FallingEdge
 - Camera Enumerations, [88](#)
- EncoderResetActivation_LevelHigh
 - Camera Enumerations, [88](#)
- EncoderResetActivation_LevelLow
 - Camera Enumerations, [88](#)
- EncoderResetActivation_RisingEdge
 - Camera Enumerations, [88](#)
- EncoderResetSource
 - quickSpin, [205](#)
- EncoderResetSource_AcquisitionEnd
 - Camera Enumerations, [88](#)
- EncoderResetSource_AcquisitionStart
 - Camera Enumerations, [88](#)
- EncoderResetSource_AcquisitionTrigger
 - Camera Enumerations, [88](#)
- EncoderResetSource_Action0
 - Camera Enumerations, [89](#)
- EncoderResetSource_Action1
 - Camera Enumerations, [89](#)
- EncoderResetSource_Action2
 - Camera Enumerations, [89](#)
- EncoderResetSource_Counter0End
 - Camera Enumerations, [89](#)
- EncoderResetSource_Counter0Start
 - Camera Enumerations, [89](#)
- EncoderResetSource_Counter1End
 - Camera Enumerations, [89](#)
- EncoderResetSource_Counter1Start
 - Camera Enumerations, [89](#)
- EncoderResetSource_Counter2End
 - Camera Enumerations, [89](#)
- EncoderResetSource_Counter2Start
 - Camera Enumerations, [89](#)
- EncoderResetSource_ExposureEnd
 - Camera Enumerations, [89](#)
- EncoderResetSource_ExposureStart
 - Camera Enumerations, [88](#)
- EncoderResetSource_FrameEnd
 - Camera Enumerations, [88](#)
- EncoderResetSource_FrameStart
 - Camera Enumerations, [88](#)
- EncoderResetSource_FrameTrigger
 - Camera Enumerations, [88](#)
- EncoderResetSource_Line0
 - Camera Enumerations, [89](#)
- EncoderResetSource_Line1

- Camera Enumerations, [89](#)
- EncoderResetSource_Line2
 - Camera Enumerations, [89](#)
- EncoderResetSource_LinkTrigger0
 - Camera Enumerations, [89](#)
- EncoderResetSource_LinkTrigger1
 - Camera Enumerations, [89](#)
- EncoderResetSource_LinkTrigger2
 - Camera Enumerations, [89](#)
- EncoderResetSource_Off
 - Camera Enumerations, [88](#)
- EncoderResetSource_SoftwareSignal0
 - Camera Enumerations, [89](#)
- EncoderResetSource_SoftwareSignal1
 - Camera Enumerations, [89](#)
- EncoderResetSource_SoftwareSignal2
 - Camera Enumerations, [89](#)
- EncoderResetSource_Timer0End
 - Camera Enumerations, [89](#)
- EncoderResetSource_Timer0Start
 - Camera Enumerations, [89](#)
- EncoderResetSource_Timer1End
 - Camera Enumerations, [89](#)
- EncoderResetSource_Timer1Start
 - Camera Enumerations, [89](#)
- EncoderResetSource_Timer2End
 - Camera Enumerations, [89](#)
- EncoderResetSource_Timer2Start
 - Camera Enumerations, [89](#)
- EncoderResetSource_UserOutput0
 - Camera Enumerations, [89](#)
- EncoderResetSource_UserOutput1
 - Camera Enumerations, [89](#)
- EncoderResetSource_UserOutput2
 - Camera Enumerations, [89](#)
- EncoderSelector
 - quickSpin, [205](#)
- EncoderSelector_Encoder0
 - Camera Enumerations, [89](#)
- EncoderSelector_Encoder1
 - Camera Enumerations, [89](#)
- EncoderSelector_Encoder2
 - Camera Enumerations, [89](#)
- EncoderSourceA
 - quickSpin, [205](#)
- EncoderSourceA_Line0
 - Camera Enumerations, [90](#)
- EncoderSourceA_Line1
 - Camera Enumerations, [90](#)
- EncoderSourceA_Line2
 - Camera Enumerations, [90](#)
- EncoderSourceA_Off
 - Camera Enumerations, [90](#)
- EncoderSourceB
 - quickSpin, [205](#)
- EncoderSourceB_Line0
 - Camera Enumerations, [90](#)
- EncoderSourceB_Line1

- Camera Enumerations, [90](#)
- EncoderSourceB_Line2
 - Camera Enumerations, [90](#)
- EncoderSourceB_Off
 - Camera Enumerations, [90](#)
- EncoderStatus
 - quickSpin, [205](#)
- EncoderStatus_EncoderDown
 - Camera Enumerations, [90](#)
- EncoderStatus_EncoderIdle
 - Camera Enumerations, [90](#)
- EncoderStatus_EncoderStatic
 - Camera Enumerations, [90](#)
- EncoderStatus_EncoderUp
 - Camera Enumerations, [90](#)
- EncoderTimeout
 - quickSpin, [205](#)
- EncoderValue
 - quickSpin, [205](#)
- EncoderValueAtReset
 - quickSpin, [205](#)
- EnumEntryNode
 - SpinnakerGenApiDefsC.h, [511](#)
- EnumerateGen2Cameras
 - quickSpinTLSystem, [280](#)
- EnumerateGEVInterfaces
 - quickSpinTLSystem, [280](#)
- EnumerateUSBInterfaces
 - quickSpinTLSystem, [281](#)
- EnumerationCount
 - quickSpin, [206](#)
- EnumerationNode
 - SpinnakerGenApiDefsC.h, [511](#)
- Error Handling, [142](#)
- Event Access, [147](#)
- EventAcquisitionEnd
 - quickSpin, [206](#)
- EventAcquisitionEndFrameID
 - quickSpin, [206](#)
- EventAcquisitionEndTimestamp
 - quickSpin, [206](#)
- EventAcquisitionError
 - quickSpin, [206](#)
- EventAcquisitionErrorFrameID
 - quickSpin, [206](#)
- EventAcquisitionErrorTimestamp
 - quickSpin, [206](#)
- EventAcquisitionStart
 - quickSpin, [206](#)
- EventAcquisitionStartFrameID
 - quickSpin, [207](#)
- EventAcquisitionStartTimestamp
 - quickSpin, [207](#)
- EventAcquisitionTransferEnd
 - quickSpin, [207](#)
- EventAcquisitionTransferEndFrameID
 - quickSpin, [207](#)
- EventAcquisitionTransferEndTimestamp
 - quickSpin, [207](#)
- EventAcquisitionTransferStart
 - quickSpin, [207](#)
- EventAcquisitionTransferStartFrameID
 - quickSpin, [207](#)
- EventAcquisitionTransferStartTimestamp
 - quickSpin, [207](#)
- EventAcquisitionTrigger
 - quickSpin, [208](#)
- EventAcquisitionTriggerFrameID
 - quickSpin, [208](#)
- EventAcquisitionTriggerTimestamp
 - quickSpin, [208](#)
- EventActionLate
 - quickSpin, [208](#)
- EventActionLateFrameID
 - quickSpin, [208](#)
- EventActionLateTimestamp
 - quickSpin, [208](#)
- EventCounter0End
 - quickSpin, [208](#)
- EventCounter0EndFrameID
 - quickSpin, [208](#)
- EventCounter0EndTimestamp
 - quickSpin, [209](#)
- EventCounter0Start
 - quickSpin, [209](#)
- EventCounter0StartFrameID
 - quickSpin, [209](#)
- EventCounter0StartTimestamp
 - quickSpin, [209](#)
- EventCounter1End
 - quickSpin, [209](#)
- EventCounter1EndFrameID
 - quickSpin, [209](#)
- EventCounter1EndTimestamp
 - quickSpin, [209](#)
- EventCounter1Start
 - quickSpin, [209](#)
- EventCounter1StartFrameID
 - quickSpin, [210](#)
- EventCounter1StartTimestamp
 - quickSpin, [210](#)
- EventEncoder0Restarted
 - quickSpin, [210](#)
- EventEncoder0RestartedFrameID
 - quickSpin, [210](#)
- EventEncoder0RestartedTimestamp
 - quickSpin, [210](#)
- EventEncoder0Stopped
 - quickSpin, [210](#)
- EventEncoder0StoppedFrameID
 - quickSpin, [210](#)
- EventEncoder0StoppedTimestamp
 - quickSpin, [210](#)
- EventEncoder1Restarted
 - quickSpin, [211](#)
- EventEncoder1RestartedFrameID

- quickSpin, [211](#)
- EventEncoder1RestartedTimestamp
 - quickSpin, [211](#)
- EventEncoder1Stopped
 - quickSpin, [211](#)
- EventEncoder1StoppedFrameID
 - quickSpin, [211](#)
- EventEncoder1StoppedTimestamp
 - quickSpin, [211](#)
- EventError
 - quickSpin, [211](#)
- EventErrorCode
 - quickSpin, [211](#)
- EventErrorFrameID
 - quickSpin, [212](#)
- EventErrorTimestamp
 - quickSpin, [212](#)
- EventExposureEnd
 - quickSpin, [212](#)
- EventExposureEndFrameID
 - quickSpin, [212](#)
- EventExposureEndTimestamp
 - quickSpin, [212](#)
- EventExposureStart
 - quickSpin, [212](#)
- EventExposureStartFrameID
 - quickSpin, [212](#)
- EventExposureStartTimestamp
 - quickSpin, [212](#)
- EventFrameBurstEnd
 - quickSpin, [213](#)
- EventFrameBurstEndFrameID
 - quickSpin, [213](#)
- EventFrameBurstEndTimestamp
 - quickSpin, [213](#)
- EventFrameBurstStart
 - quickSpin, [213](#)
- EventFrameBurstStartFrameID
 - quickSpin, [213](#)
- EventFrameBurstStartTimestamp
 - quickSpin, [213](#)
- EventFrameEnd
 - quickSpin, [213](#)
- EventFrameEndFrameID
 - quickSpin, [213](#)
- EventFrameEndTimestamp
 - quickSpin, [214](#)
- EventFrameStart
 - quickSpin, [214](#)
- EventFrameStartFrameID
 - quickSpin, [214](#)
- EventFrameStartTimestamp
 - quickSpin, [214](#)
- EventFrameTransferEnd
 - quickSpin, [214](#)
- EventFrameTransferEndFrameID
 - quickSpin, [214](#)
- EventFrameTransferEndTimestamp
 - quickSpin, [214](#)
- EventFrameTransferStart
 - quickSpin, [214](#)
- EventFrameTransferStartFrameID
 - quickSpin, [215](#)
- EventFrameTransferStartTimestamp
 - quickSpin, [215](#)
- EventFrameTrigger
 - quickSpin, [215](#)
- EventFrameTriggerFrameID
 - quickSpin, [215](#)
- EventFrameTriggerTimestamp
 - quickSpin, [215](#)
- EventLine0AnyEdge
 - quickSpin, [215](#)
- EventLine0AnyEdgeFrameID
 - quickSpin, [215](#)
- EventLine0AnyEdgeTimestamp
 - quickSpin, [215](#)
- EventLine0FallingEdge
 - quickSpin, [216](#)
- EventLine0FallingEdgeFrameID
 - quickSpin, [216](#)
- EventLine0FallingEdgeTimestamp
 - quickSpin, [216](#)
- EventLine0RisingEdge
 - quickSpin, [216](#)
- EventLine0RisingEdgeFrameID
 - quickSpin, [216](#)
- EventLine0RisingEdgeTimestamp
 - quickSpin, [216](#)
- EventLine1AnyEdge
 - quickSpin, [216](#)
- EventLine1AnyEdgeFrameID
 - quickSpin, [216](#)
- EventLine1AnyEdgeTimestamp
 - quickSpin, [217](#)
- EventLine1FallingEdge
 - quickSpin, [217](#)
- EventLine1FallingEdgeFrameID
 - quickSpin, [217](#)
- EventLine1FallingEdgeTimestamp
 - quickSpin, [217](#)
- EventLine1RisingEdge
 - quickSpin, [217](#)
- EventLine1RisingEdgeFrameID
 - quickSpin, [217](#)
- EventLine1RisingEdgeTimestamp
 - quickSpin, [217](#)
- EventLinkSpeedChange
 - quickSpin, [217](#)
- EventLinkSpeedChangeFrameID
 - quickSpin, [218](#)
- EventLinkSpeedChangeTimestamp
 - quickSpin, [218](#)
- EventLinkTrigger0
 - quickSpin, [218](#)
- EventLinkTrigger0FrameID

- quickSpin, [218](#)
- EventLinkTrigger0Timestamp
 - quickSpin, [218](#)
- EventLinkTrigger1
 - quickSpin, [218](#)
- EventLinkTrigger1FrameID
 - quickSpin, [218](#)
- EventLinkTrigger1Timestamp
 - quickSpin, [218](#)
- EventNotification
 - quickSpin, [219](#)
- EventNotification_Off
 - Camera Enumerations, [91](#)
- EventNotification_On
 - Camera Enumerations, [91](#)
- EventSelector
 - quickSpin, [219](#)
- EventSelector_Error
 - Camera Enumerations, [91](#)
- EventSelector_ExposureEnd
 - Camera Enumerations, [91](#)
- EventSelector_SerialPortReceive
 - Camera Enumerations, [91](#)
- EventSequencerSetChange
 - quickSpin, [219](#)
- EventSequencerSetChangeFrameID
 - quickSpin, [219](#)
- EventSequencerSetChangeTimestamp
 - quickSpin, [219](#)
- EventSerialData
 - quickSpin, [219](#)
- EventSerialDataLength
 - quickSpin, [219](#)
- EventSerialPortReceive
 - quickSpin, [219](#)
- EventSerialPortReceiveTimestamp
 - quickSpin, [220](#)
- EventSerialReceiveOverflow
 - quickSpin, [220](#)
- EventStream0TransferBlockEnd
 - quickSpin, [220](#)
- EventStream0TransferBlockEndFrameID
 - quickSpin, [220](#)
- EventStream0TransferBlockEndTimestamp
 - quickSpin, [220](#)
- EventStream0TransferBlockStart
 - quickSpin, [220](#)
- EventStream0TransferBlockStartFrameID
 - quickSpin, [220](#)
- EventStream0TransferBlockStartTimestamp
 - quickSpin, [220](#)
- EventStream0TransferBlockTrigger
 - quickSpin, [221](#)
- EventStream0TransferBlockTriggerFrameID
 - quickSpin, [221](#)
- EventStream0TransferBlockTriggerTimestamp
 - quickSpin, [221](#)
- EventStream0TransferBurstEnd
 - quickSpin, [221](#)
- EventStream0TransferBurstEndFrameID
 - quickSpin, [221](#)
- EventStream0TransferBurstEndTimestamp
 - quickSpin, [221](#)
- EventStream0TransferBurstStart
 - quickSpin, [221](#)
- EventStream0TransferBurstStartFrameID
 - quickSpin, [221](#)
- EventStream0TransferBurstStartTimestamp
 - quickSpin, [222](#)
- EventStream0TransferEnd
 - quickSpin, [222](#)
- EventStream0TransferEndFrameID
 - quickSpin, [222](#)
- EventStream0TransferEndTimestamp
 - quickSpin, [222](#)
- EventStream0TransferOverflow
 - quickSpin, [222](#)
- EventStream0TransferOverflowFrameID
 - quickSpin, [222](#)
- EventStream0TransferOverflowTimestamp
 - quickSpin, [222](#)
- EventStream0TransferPause
 - quickSpin, [222](#)
- EventStream0TransferPauseFrameID
 - quickSpin, [223](#)
- EventStream0TransferPauseTimestamp
 - quickSpin, [223](#)
- EventStream0TransferResume
 - quickSpin, [223](#)
- EventStream0TransferResumeFrameID
 - quickSpin, [223](#)
- EventStream0TransferResumeTimestamp
 - quickSpin, [223](#)
- EventStream0TransferStart
 - quickSpin, [223](#)
- EventStream0TransferStartFrameID
 - quickSpin, [223](#)
- EventStream0TransferStartTimestamp
 - quickSpin, [223](#)
- EventTest
 - quickSpin, [224](#)
- EventTestTimestamp
 - quickSpin, [224](#)
- EventTimer0End
 - quickSpin, [224](#)
- EventTimer0EndFrameID
 - quickSpin, [224](#)
- EventTimer0EndTimestamp
 - quickSpin, [224](#)
- EventTimer0Start
 - quickSpin, [224](#)
- EventTimer0StartFrameID
 - quickSpin, [224](#)
- EventTimer0StartTimestamp
 - quickSpin, [224](#)
- EventTimer1End

- quickSpin, [225](#)
- EventTimer1 EndFrameID
 - quickSpin, [225](#)
- EventTimer1 EndTimestamp
 - quickSpin, [225](#)
- EventTimer1 Start
 - quickSpin, [225](#)
- EventTimer1 StartFrameID
 - quickSpin, [225](#)
- EventTimer1 StartTimestamp
 - quickSpin, [225](#)
- Expert
 - SpinnakerGenApiDefsC.h, [512](#)
- ExposureActiveMode
 - quickSpin, [225](#)
- ExposureActiveMode_AllPixels
 - Camera Enumerations, [91](#)
- ExposureActiveMode_AnyPixels
 - Camera Enumerations, [91](#)
- ExposureActiveMode_Line1
 - Camera Enumerations, [91](#)
- ExposureAuto
 - quickSpin, [225](#)
- ExposureAuto_Continuous
 - Camera Enumerations, [92](#)
- ExposureAuto_Off
 - Camera Enumerations, [92](#)
- ExposureAuto_Once
 - Camera Enumerations, [92](#)
- ExposureMode
 - quickSpin, [226](#)
- ExposureMode_Timed
 - Camera Enumerations, [92](#)
- ExposureMode_TriggerWidth
 - Camera Enumerations, [92](#)
- ExposureTime
 - quickSpin, [226](#)
- ExposureTimeMode
 - quickSpin, [226](#)
- ExposureTimeMode_Common
 - Camera Enumerations, [92](#)
- ExposureTimeMode_Individual
 - Camera Enumerations, [92](#)
- ExposureTimeSelector
 - quickSpin, [226](#)
- ExposureTimeSelector_Blue
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Common
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Cyan
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Green
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Infrared
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Magenta
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Red
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Stage1
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Stage2
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Ultraviolet
 - Camera Enumerations, [93](#)
- ExposureTimeSelector_Yellow
 - Camera Enumerations, [93](#)
- FactoryReset
 - quickSpin, [226](#)
- False
 - SpinnakerDefsC.h, [462](#)
- FileAccessBuffer
 - quickSpin, [226](#)
- FileAccessLength
 - quickSpin, [226](#)
- FileAccessOffset
 - quickSpin, [226](#)
- FileOpenMode
 - quickSpin, [227](#)
- FileOpenMode_Read
 - Camera Enumerations, [93](#)
- FileOpenMode_ReadWrite
 - Camera Enumerations, [93](#)
- FileOpenMode_Write
 - Camera Enumerations, [93](#)
- FileOperationExecute
 - quickSpin, [227](#)
- FileOperationResult
 - quickSpin, [227](#)
- FileOperationSelector
 - quickSpin, [227](#)
- FileOperationSelector_Close
 - Camera Enumerations, [94](#)
- FileOperationSelector_Delete
 - Camera Enumerations, [94](#)
- FileOperationSelector_Open
 - Camera Enumerations, [94](#)
- FileOperationSelector_Read
 - Camera Enumerations, [94](#)
- FileOperationSelector_Write
 - Camera Enumerations, [94](#)
- FileOperationStatus
 - quickSpin, [227](#)
- FileOperationStatus_Failure
 - Camera Enumerations, [94](#)
- FileOperationStatus_Overflow
 - Camera Enumerations, [94](#)
- FileOperationStatus_Success
 - Camera Enumerations, [94](#)
- FileSelector
 - quickSpin, [227](#)
- FileSelector_SerialPort0
 - Camera Enumerations, [94](#)
- FileSelector_UserFile1
 - Camera Enumerations, [94](#)
- FileSelector_UserSet0

- Camera Enumerations, [94](#)
- FileSelector_UserSet1
 - Camera Enumerations, [94](#)
- FileSelector_UserSetDefault
 - Camera Enumerations, [94](#)
- FileSize
 - quickSpin, [227](#)
- fixedIncrement
 - SpinnakerGenApiDefsC.h, [508](#)
- FLIRFilterDriverStatus
 - quickSpinTLInterface, [267](#)
- FLIRFilterDriverStatus_Disabled
 - Transport Layer Enumerations, [155](#)
- FLIRFilterDriverStatus_Enabled
 - Transport Layer Enumerations, [155](#)
- FLIRFilterDriverStatus_NotSupported
 - Transport Layer Enumerations, [155](#)
- FloatNode
 - SpinnakerGenApiDefsC.h, [510](#)
- fnAutomatic
 - SpinnakerGenApiDefsC.h, [507](#)
- fnFixed
 - SpinnakerGenApiDefsC.h, [507](#)
- fnScientific
 - SpinnakerGenApiDefsC.h, [507](#)
- frameRate
 - spinAVIOption, [284](#)
 - spinH264Option, [293](#)
 - spinMJPEGOption, [297](#)
- Gain
 - quickSpin, [227](#)
- GainAuto
 - quickSpin, [228](#)
- GainAuto_Continuous
 - Camera Enumerations, [96](#)
- GainAuto_Off
 - Camera Enumerations, [96](#)
- GainAuto_Once
 - Camera Enumerations, [96](#)
- GainAutoBalance
 - quickSpin, [228](#)
- GainAutoBalance_Continuous
 - Camera Enumerations, [96](#)
- GainAutoBalance_Off
 - Camera Enumerations, [96](#)
- GainAutoBalance_Once
 - Camera Enumerations, [96](#)
- GainSelector
 - quickSpin, [228](#)
- GainSelector_All
 - Camera Enumerations, [96](#)
- Gamma
 - quickSpin, [228](#)
- GammaEnable
 - quickSpin, [228](#)
- GenICamXMLLocation
 - quickSpinTLDevice, [262](#)
- GenICamXMLLocation_Device
 - Transport Layer Enumerations, [155](#)
- GenICamXMLLocation_Host
 - Transport Layer Enumerations, [155](#)
- GenICamXMLPath
 - quickSpinTLDevice, [262](#)
- GenTLSFNCVersionMajor
 - quickSpinTLSystem, [281](#)
- GenTLSFNCVersionMinor
 - quickSpinTLSystem, [281](#)
- GenTLSFNCVersionSubMinor
 - quickSpinTLSystem, [281](#)
- GenTLVersionMajor
 - quickSpinTLSystem, [281](#)
- GenTLVersionMinor
 - quickSpinTLSystem, [281](#)
- GEV
 - SpinnakerGenApiDefsC.h, [512](#)
- GevActionAckRequired
 - quickSpinTLInterface, [268](#)
- GevActionDeviceKey
 - quickSpinTLInterface, [268](#)
- GevActionGroupKey
 - quickSpinTLInterface, [268](#)
- GevActionGroupMask
 - quickSpinTLInterface, [268](#)
- GevActionTime
 - quickSpinTLInterface, [268](#)
- GevActiveLinkCount
 - quickSpin, [228](#)
- GevAutoAssignIPEnable
 - quickSpinTLSystem, [281](#)
- GevCCP
 - quickSpin, [228](#)
 - quickSpinTLDevice, [262](#)
- GevCCP_ControlAccess
 - Camera Enumerations, [97](#)
- GevCCP_EnumEntry_GevCCP_ControlAccess
 - Transport Layer Enumerations, [155](#)
- GevCCP_EnumEntry_GevCCP_ExclusiveAccess
 - Transport Layer Enumerations, [155](#)
- GevCCP_EnumEntry_GevCCP_OpenAccess
 - Transport Layer Enumerations, [155](#)
- GevCCP_ExclusiveAccess
 - Camera Enumerations, [97](#)
- GevCCP_OpenAccess
 - Camera Enumerations, [97](#)
- GevCurrentDefaultGateway
 - quickSpin, [228](#)
- GevCurrentIPAddress
 - quickSpin, [229](#)
- GevCurrentIPConfigurationDHCP
 - quickSpin, [229](#)
- GevCurrentIPConfigurationLLA
 - quickSpin, [229](#)
- GevCurrentIPConfigurationPersistentIP
 - quickSpin, [229](#)
- GevCurrentPhysicalLinkConfiguration
 - quickSpin, [229](#)

- GevCurrentPhysicalLinkConfiguration_DynamicLAG
 - Camera Enumerations, [97](#)
- GevCurrentPhysicalLinkConfiguration_MultiLink
 - Camera Enumerations, [97](#)
- GevCurrentPhysicalLinkConfiguration_SingleLink
 - Camera Enumerations, [97](#)
- GevCurrentPhysicalLinkConfiguration_StaticLAG
 - Camera Enumerations, [97](#)
- GevCurrentSubnetMask
 - quickSpin, [229](#)
- GevDeviceAutoForceIP
 - quickSpinTLDevice, [262](#)
 - quickSpinTLInterface, [268](#)
- GevDeviceDisableDiscovery
 - quickSpinTLInterface, [268](#)
- GevDeviceDiscoverMaximumPacketSize
 - quickSpinTLDevice, [263](#)
- GevDeviceDiscoveryEnabled
 - quickSpinTLInterface, [268](#)
- GevDeviceEnableDiscovery
 - quickSpinTLInterface, [269](#)
- GevDeviceForceGateway
 - quickSpinTLDevice, [263](#)
 - quickSpinTLInterface, [269](#)
- GevDeviceForceIP
 - quickSpinTLDevice, [263](#)
 - quickSpinTLInterface, [269](#)
- GevDeviceForceIPAddress
 - quickSpinTLDevice, [263](#)
 - quickSpinTLInterface, [269](#)
- GevDeviceForceSubnetMask
 - quickSpinTLDevice, [263](#)
 - quickSpinTLInterface, [269](#)
- GevDeviceGateway
 - quickSpinTLDevice, [263](#)
 - quickSpinTLInterface, [269](#)
- GevDeviceIPAddress
 - quickSpinTLDevice, [263](#)
 - quickSpinTLInterface, [269](#)
- GevDevicesWrongSubnet
 - quickSpinTLDevice, [263](#)
- GevDeviceMACAddress
 - quickSpinTLDevice, [264](#)
 - quickSpinTLInterface, [269](#)
- GevDeviceMaximumPacketSize
 - quickSpinTLDevice, [264](#)
- GevDeviceMaximumRetryCount
 - quickSpinTLDevice, [264](#)
- GevDeviceModelsBigEndian
 - quickSpinTLDevice, [264](#)
- GevDevicePort
 - quickSpinTLDevice, [264](#)
- GevDeviceReadAndWriteTimeout
 - quickSpinTLDevice, [264](#)
- GevDeviceSubnetMask
 - quickSpinTLDevice, [264](#)
 - quickSpinTLInterface, [270](#)
- GevDiscoveryAckDelay
 - quickSpin, [229](#)
- GevFirstURL
 - quickSpin, [229](#)
- GevGVCPExtendedStatusCodes
 - quickSpin, [230](#)
- GevGVCPExtendedStatusCodesSelector
 - quickSpin, [230](#)
- GevGVCPExtendedStatusCodesSelector_Version1_1
 - Camera Enumerations, [97](#)
- GevGVCPExtendedStatusCodesSelector_Version2_0
 - Camera Enumerations, [97](#)
- GevGVCPHeartbeatDisable
 - quickSpin, [230](#)
- GevGVCPPendingAck
 - quickSpin, [230](#)
- GevGVCPPendingTimeout
 - quickSpin, [230](#)
- GevGVSPExtendedIDMode
 - quickSpin, [230](#)
- GevGVSPExtendedIDMode_Off
 - Camera Enumerations, [98](#)
- GevGVSPExtendedIDMode_On
 - Camera Enumerations, [98](#)
- GevHeartbeatTimeout
 - quickSpin, [230](#)
- GevIEEE1588
 - quickSpin, [230](#)
- GevIEEE1588ClockAccuracy
 - quickSpin, [231](#)
- GevIEEE1588ClockAccuracy_Unknown
 - Camera Enumerations, [98](#)
- GevIEEE1588Mode
 - quickSpin, [231](#)
- GevIEEE1588Mode_Auto
 - Camera Enumerations, [98](#)
- GevIEEE1588Mode_SlaveOnly
 - Camera Enumerations, [98](#)
- GevIEEE1588Status
 - quickSpin, [231](#)
- GevIEEE1588Status_Disabled
 - Camera Enumerations, [98](#)
- GevIEEE1588Status_Faulty
 - Camera Enumerations, [98](#)
- GevIEEE1588Status_Initializing
 - Camera Enumerations, [98](#)
- GevIEEE1588Status_Listening
 - Camera Enumerations, [99](#)
- GevIEEE1588Status_Master
 - Camera Enumerations, [99](#)
- GevIEEE1588Status_Passive
 - Camera Enumerations, [99](#)
- GevIEEE1588Status_PreMaster
 - Camera Enumerations, [99](#)
- GevIEEE1588Status_Slave
 - Camera Enumerations, [99](#)
- GevIEEE1588Status_Uncalibrated
 - Camera Enumerations, [99](#)
- GevInterfaceDefaultGateway

- quickSpinTLSystem, [281](#)
- GevInterfaceDefaultIPAddress
 - quickSpinTLSystem, [282](#)
- GevInterfaceDefaultSubnetMask
 - quickSpinTLSystem, [282](#)
- GevInterfaceGateway
 - quickSpinTLInterface, [270](#)
- GevInterfaceGatewaySelector
 - quickSpinTLInterface, [270](#)
- GevInterfaceIsIPConflict
 - quickSpinTLInterface, [270](#)
- GevInterfaceMACAddress
 - quickSpinTLInterface, [270](#)
 - quickSpinTLSystem, [282](#)
- GevInterfaceMTU
 - quickSpinTLInterface, [270](#)
- GevInterfaceReceiveLinkSpeed
 - quickSpinTLInterface, [270](#)
- GevInterfaceSelector
 - quickSpin, [231](#)
- GevInterfaceSubnetIPAddress
 - quickSpinTLInterface, [270](#)
- GevInterfaceSubnetMask
 - quickSpinTLInterface, [271](#)
- GevInterfaceSubnetSelector
 - quickSpinTLInterface, [271](#)
- GevInterfaceTransmitLinkSpeed
 - quickSpinTLInterface, [271](#)
- GevIPConfigurationStatus
 - quickSpin, [231](#)
- GevIPConfigurationStatus_DHCP
 - Camera Enumerations, [99](#)
- GevIPConfigurationStatus_ForceIP
 - Camera Enumerations, [99](#)
- GevIPConfigurationStatus_LLA
 - Camera Enumerations, [99](#)
- GevIPConfigurationStatus_None
 - Camera Enumerations, [99](#)
- GevIPConfigurationStatus_PersistentIP
 - Camera Enumerations, [99](#)
- GevMACAddress
 - quickSpin, [231](#)
- GevMCDA
 - quickSpin, [231](#)
- GevMCPHostPort
 - quickSpin, [231](#)
- GevMCRC
 - quickSpin, [232](#)
- GevMCSP
 - quickSpin, [232](#)
- GevMCTT
 - quickSpin, [232](#)
- GevNumberOfInterfaces
 - quickSpin, [232](#)
- GevPAUSEFrameReception
 - quickSpin, [232](#)
- GevPAUSEFrameTransmission
 - quickSpin, [232](#)
- GevPersistentDefaultGateway
 - quickSpin, [232](#)
- GevPersistentIPAddress
 - quickSpin, [232](#)
- GevPersistentSubnetMask
 - quickSpin, [233](#)
- GevPhysicalLinkConfiguration
 - quickSpin, [233](#)
- GevPhysicalLinkConfiguration_DynamicLAG
 - Camera Enumerations, [99](#)
- GevPhysicalLinkConfiguration_MultiLink
 - Camera Enumerations, [99](#)
- GevPhysicalLinkConfiguration_SingleLink
 - Camera Enumerations, [99](#)
- GevPhysicalLinkConfiguration_StaticLAG
 - Camera Enumerations, [99](#)
- GevPrimaryApplicationIPAddress
 - quickSpin, [233](#)
- GevPrimaryApplicationSocket
 - quickSpin, [233](#)
- GevPrimaryApplicationSwitchoverKey
 - quickSpin, [233](#)
- GevSCCFGAllInTransmission
 - quickSpin, [233](#)
- GevSCCFGExtendedChunkData
 - quickSpin, [233](#)
- GevSCCFGPacketResendDestination
 - quickSpin, [233](#)
- GevSCCFGUnconditionalStreaming
 - quickSpin, [234](#)
- GevSCDA
 - quickSpin, [234](#)
- GevSCPD
 - quickSpin, [234](#)
- GevSCPDDirection
 - quickSpin, [234](#)
- GevSCPHostPort
 - quickSpin, [234](#)
- GevSCPIInterfaceIndex
 - quickSpin, [234](#)
- GevSCPSBigEndian
 - quickSpin, [234](#)
- GevSCPSDoNotFragment
 - quickSpin, [234](#)
- GevSCPSFireTestPacket
 - quickSpin, [235](#)
- GevSCPSPacketSize
 - quickSpin, [235](#)
- GevSCSP
 - quickSpin, [235](#)
- GevSCZoneConfigurationLock
 - quickSpin, [235](#)
- GevSCZoneCount
 - quickSpin, [235](#)
- GevSCZoneDirectionAll
 - quickSpin, [235](#)
- GevSecondURL
 - quickSpin, [235](#)

- GevStreamChannelSelector
 - quickSpin, [235](#)
- GevSupportedOption
 - quickSpin, [236](#)
- GevSupportedOptionSelector
 - quickSpin, [236](#)
- GevSupportedOptionSelector_Action
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_CCPApplicationSocket
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_CommandsConcatenation
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_DiscoveryAckDelay
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_DiscoveryAckDelayWritable
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_Event
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_EventData
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_ExtendedStatusCodes
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_HeartbeatDisable
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_IPConfigurationDHCP
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_IPConfigurationLLA
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_IPConfigurationPersistentIP
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_LinkSpeed
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_ManifestTable
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_MessageChannelSourceSocket
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_PacketResend
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_PendingAck
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_SerialNumber
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_StreamChannelSourceSocket
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_TestData
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_UserDefinedName
 - Camera Enumerations, [100](#)
- GevSupportedOptionSelector_WriteMem
 - Camera Enumerations, [100](#)
- GevTimestampTickFrequency
 - quickSpin, [236](#)
- GevVersionMajor
 - quickSpinTLDevice, [264](#)
 - quickSpinTLSystem, [282](#)
- GevVersionMinor
 - quickSpinTLDevice, [265](#)
 - quickSpinTLSystem, [282](#)
- GUIXMLLocation
 - quickSpinTLDevice, [265](#)
- GUIXMLLocation_Device
 - Transport Layer Enumerations, [156](#)
- GUIXMLLocation_Host
 - Transport Layer Enumerations, [156](#)
- GuiXmlManifestAddress
 - quickSpin, [236](#)
- GUIXMLPath
 - quickSpinTLDevice, [265](#)
- Guru
 - SpinnakerGenApiDefsC.h, [512](#)
- Height
 - quickSpin, [236](#)
- height
 - spinAVIOption, [284](#)
 - spinH264Option, [293](#)
 - spinMJPEGOption, [297](#)
- HeightMax
 - quickSpin, [236](#)
- HexNumber
 - SpinnakerGenApiDefsC.h, [511](#)
- HostAdapterDriverVersion
 - quickSpinTLInterface, [271](#)
- HostAdapterName
 - quickSpinTLInterface, [271](#)
- HostAdapterVendor
 - quickSpinTLInterface, [271](#)
- IBoolean Access, [150](#)
- ICategory Access, [150](#)
- ICommand Access, [150](#)
- idFrom
 - SpinnakerGenApiDefsC.h, [508](#)
- idNone
 - SpinnakerGenApiDefsC.h, [508](#)
- idTo
 - SpinnakerGenApiDefsC.h, [508](#)
- IEnumEntry Access, [150](#)
- IEnumeration Access, [149](#)
- IFloat Access, [149](#)
- IPC
 - SpinnakerGenApiDefsC.h, [512](#)
- IInteger Access, [149](#)
- Image Access, [144](#)
- Image Processor Access, [144](#)
- ImageComponentEnable
 - quickSpin, [236](#)
- ImageComponentSelector
 - quickSpin, [236](#)
- ImageComponentSelector_Color
 - Camera Enumerations, [100](#)
- ImageComponentSelector_Confidence
 - Camera Enumerations, [101](#)
- ImageComponentSelector_Disparity
 - Camera Enumerations, [101](#)
- ImageComponentSelector_Infrared
 - Camera Enumerations, [100](#)

- ImageComponentSelector_Intensity
 - Camera Enumerations, [100](#)
- ImageComponentSelector_Range
 - Camera Enumerations, [100](#)
- ImageComponentSelector_Scatter
 - Camera Enumerations, [101](#)
- ImageComponentSelector_Ultraviolet
 - Camera Enumerations, [100](#)
- ImageCompressionBitrate
 - quickSpin, [237](#)
- ImageCompressionJPEGFormatOption
 - quickSpin, [237](#)
- ImageCompressionJPEGFormatOption_BaselineOptimized
 - Camera Enumerations, [101](#)
- ImageCompressionJPEGFormatOption_BaselineStandard
 - Camera Enumerations, [101](#)
- ImageCompressionJPEGFormatOption_Lossless
 - Camera Enumerations, [101](#)
- ImageCompressionJPEGFormatOption_Progressive
 - Camera Enumerations, [101](#)
- ImageCompressionMode
 - quickSpin, [237](#)
- ImageCompressionMode_Lossless
 - Camera Enumerations, [102](#)
- ImageCompressionMode_Off
 - Camera Enumerations, [102](#)
- ImageCompressionQuality
 - quickSpin, [237](#)
- ImageCompressionRateOption
 - quickSpin, [237](#)
- ImageCompressionRateOption_FixBitrate
 - Camera Enumerations, [102](#)
- ImageCompressionRateOption_FixQuality
 - Camera Enumerations, [102](#)
- ImageList Access, [143](#)
- ImageStatistics Access, [147](#)
- include/spinc/CameraDefsC.h, [303](#)
- include/spinc/ChunkDataDefC.h, [336](#)
- include/spinc/QuickSpinC.h, [337](#)
- include/spinc/QuickSpinDefsC.h, [339](#)
- include/spinc/SpinnakerC.h, [341](#)
- include/spinc/SpinnakerDefsC.h, [447](#)
- include/spinc/SpinnakerGenApiC.h, [462](#)
- include/spinc/SpinnakerGenApiDefsC.h, [503](#)
- include/spinc/SpinnakerPlatformC.h, [514](#)
- include/spinc/SpinVideoC.h, [515](#)
- include/spinc/TransportLayerDefsC.h, [517](#)
- include/spinc/TransportLayerDeviceC.h, [519](#)
- include/spinc/TransportLayerInterfaceC.h, [520](#)
- include/spinc/TransportLayerStreamC.h, [520](#)
- include/spinc/TransportLayerSystemC.h, [521](#)
- IncompatibleDeviceCount
 - quickSpinTLInterface, [271](#)
- IncompatibleDeviceID
 - quickSpinTLInterface, [271](#)
- IncompatibleDeviceModelName
 - quickSpinTLInterface, [272](#)
- IncompatibleDeviceSelector
 - quickSpinTLInterface, [272](#)
- IncompatibleDeviceVendorName
 - quickSpinTLInterface, [272](#)
- IncompatibleGevDeviceIPAddress
 - quickSpinTLInterface, [272](#)
- IncompatibleGevDeviceMACAddress
 - quickSpinTLInterface, [272](#)
- IncompatibleGevDeviceSubnetMask
 - quickSpinTLInterface, [272](#)
- Increasing
 - SpinnakerGenApiDefsC.h, [512](#)
- indexedColor_8bit
 - spinBMPOption, [285](#)
- IntegerNode
 - SpinnakerGenApiDefsC.h, [510](#)
- Interface Access, [143](#)
- InterfaceDisplayName
 - quickSpinTLInterface, [272](#)
 - quickSpinTLSystem, [282](#)
- InterfaceID
 - quickSpinTLInterface, [272](#)
 - quickSpinTLSystem, [282](#)
- InterfaceList Access, [143](#)
- InterfaceSelector
 - quickSpinTLSystem, [282](#)
- InterfaceType
 - quickSpinTLInterface, [273](#)
- InterfaceType_CameraLink
 - Transport Layer Enumerations, [156](#)
- InterfaceType_CameraLinkHS
 - Transport Layer Enumerations, [156](#)
- InterfaceType_CoaxPress
 - Transport Layer Enumerations, [156](#)
- InterfaceType_Custom
 - Transport Layer Enumerations, [156](#)
- InterfaceType_GigEVision
 - Transport Layer Enumerations, [156](#)
- InterfaceType_USB3Vision
 - Transport Layer Enumerations, [156](#)
- InterfaceUpdateList
 - quickSpinTLSystem, [283](#)
- interlaced
 - spinPNGOption, [300](#)
- intfIBase
 - SpinnakerGenApiDefsC.h, [509](#)
- intfIBoolean
 - SpinnakerGenApiDefsC.h, [509](#)
- intfICategory
 - SpinnakerGenApiDefsC.h, [509](#)
- intfICommand
 - SpinnakerGenApiDefsC.h, [509](#)
- intfIEnumEntry
 - SpinnakerGenApiDefsC.h, [509](#)
- intfIEnumeration
 - SpinnakerGenApiDefsC.h, [509](#)
- intfIFloat
 - SpinnakerGenApiDefsC.h, [509](#)
- intfIInteger

- SpinnakerGenApiDefsC.h, [509](#)
- intfIPort
 - SpinnakerGenApiDefsC.h, [509](#)
- intfIRegister
 - SpinnakerGenApiDefsC.h, [509](#)
- intfIString
 - SpinnakerGenApiDefsC.h, [509](#)
- intfIValue
 - SpinnakerGenApiDefsC.h, [509](#)
- Invisible
 - SpinnakerGenApiDefsC.h, [512](#)
- IPv4Address
 - SpinnakerGenApiDefsC.h, [511](#)
- IRegister Access, [150](#)
- IspEnable
 - quickSpin, [237](#)
- IValue Access, [149](#)
- Linear
 - SpinnakerGenApiDefsC.h, [511](#)
- LineFilterWidth
 - quickSpin, [237](#)
- LineFormat
 - quickSpin, [237](#)
- LineFormat_LVDS
 - Camera Enumerations, [102](#)
- LineFormat_NoConnect
 - Camera Enumerations, [102](#)
- LineFormat_OpenDrain
 - Camera Enumerations, [102](#)
- LineFormat_OptoCoupled
 - Camera Enumerations, [102](#)
- LineFormat_RS422
 - Camera Enumerations, [102](#)
- LineFormat_TriState
 - Camera Enumerations, [102](#)
- LineFormat_TTL
 - Camera Enumerations, [102](#)
- LineInputFilterSelector
 - quickSpin, [238](#)
- LineInputFilterSelector_Debounce
 - Camera Enumerations, [103](#)
- LineInputFilterSelector_Deg glitch
 - Camera Enumerations, [103](#)
- LineInverter
 - quickSpin, [238](#)
- LineMode
 - quickSpin, [238](#)
- LineMode_Input
 - Camera Enumerations, [103](#)
- LineMode_Output
 - Camera Enumerations, [103](#)
- LinePitch
 - quickSpin, [238](#)
- LineSelector
 - quickSpin, [238](#)
- LineSelector_Line0
 - Camera Enumerations, [103](#)
- LineSelector_Line1
 - Camera Enumerations, [103](#)
- LineSelector_Line2
 - Camera Enumerations, [103](#)
- LineSelector_Line3
 - Camera Enumerations, [103](#)
- LineSource
 - quickSpin, [238](#)
- LineSource_AllPixel
 - Camera Enumerations, [104](#)
- LineSource_AnyPixel
 - Camera Enumerations, [104](#)
- LineSource_Counter0Active
 - Camera Enumerations, [104](#)
- LineSource_Counter1Active
 - Camera Enumerations, [104](#)
- LineSource_ExposureActive
 - Camera Enumerations, [104](#)
- LineSource_FrameTriggerWait
 - Camera Enumerations, [104](#)
- LineSource_Line0
 - Camera Enumerations, [104](#)
- LineSource_Line1
 - Camera Enumerations, [104](#)
- LineSource_Line2
 - Camera Enumerations, [104](#)
- LineSource_Line3
 - Camera Enumerations, [104](#)
- LineSource_LogicBlock0
 - Camera Enumerations, [104](#)
- LineSource_LogicBlock1
 - Camera Enumerations, [104](#)
- LineSource_Off
 - Camera Enumerations, [104](#)
- LineSource_PPSSignal
 - Camera Enumerations, [104](#)
- LineSource_SerialPort0
 - Camera Enumerations, [104](#)
- LineSource_UserOutput0
 - Camera Enumerations, [104](#)
- LineSource_UserOutput1
 - Camera Enumerations, [104](#)
- LineSource_UserOutput2
 - Camera Enumerations, [104](#)
- LineSource_UserOutput3
 - Camera Enumerations, [104](#)
- LineStatus
 - quickSpin, [238](#)
- LineStatusAll
 - quickSpin, [238](#)
- LinkErrorCount
 - quickSpin, [239](#)
- LinkUptime
 - quickSpin, [239](#)
- listIncrement
 - SpinnakerGenApiDefsC.h, [508](#)
- LittleEndian
 - SpinnakerGenApiDefsC.h, [508](#)
- Logarithmic

- SpinnakerGenApiDefsC.h, [511](#)
- Logging Event Data Access, [147](#)
- LogicBlockLUTInputActivation
 - quickSpin, [239](#)
- LogicBlockLUTInputActivation_AnyEdge
 - Camera Enumerations, [104](#)
- LogicBlockLUTInputActivation_FallingEdge
 - Camera Enumerations, [104](#)
- LogicBlockLUTInputActivation_LevelHigh
 - Camera Enumerations, [104](#)
- LogicBlockLUTInputActivation_LevelLow
 - Camera Enumerations, [104](#)
- LogicBlockLUTInputActivation_RisingEdge
 - Camera Enumerations, [104](#)
- LogicBlockLUTInputSelector
 - quickSpin, [239](#)
- LogicBlockLUTInputSelector_Input0
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSelector_Input1
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSelector_Input2
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSelector_Input3
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource
 - quickSpin, [239](#)
- LogicBlockLUTInputSource_AcquisitionActive
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Counter0End
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Counter0Start
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Counter1End
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Counter1Start
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_ExposureEnd
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_ExposureStart
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_FrameTriggerWait
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Line0
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Line1
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Line2
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Line3
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_LogicBlock0
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_LogicBlock1
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_UserOutput0
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_UserOutput1
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_UserOutput2
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_UserOutput3
 - Camera Enumerations, [105](#)
- LogicBlockLUTInputSource_Zero
 - Camera Enumerations, [105](#)
- LogicBlockLUTOutputValue
 - quickSpin, [239](#)
- LogicBlockLUTOutputValueAll
 - quickSpin, [239](#)
- LogicBlockLUTRowIndex
 - quickSpin, [239](#)
- LogicBlockLUTSelector
 - quickSpin, [240](#)
- LogicBlockLUTSelector_Enable
 - Camera Enumerations, [106](#)
- LogicBlockLUTSelector_Value
 - Camera Enumerations, [106](#)
- LogicBlockSelector
 - quickSpin, [240](#)
- LogicBlockSelector_LogicBlock0
 - Camera Enumerations, [106](#)
- LogicBlockSelector_LogicBlock1
 - Camera Enumerations, [106](#)
- LUTEnable
 - quickSpin, [240](#)
- LUTIndex
 - quickSpin, [240](#)
- LUTSelector
 - quickSpin, [240](#)
- LUTSelector_LUT1
 - Camera Enumerations, [106](#)
- LUTValue
 - quickSpin, [240](#)
- LUTValueAll
 - quickSpin, [240](#)
- m_blackLevel
 - spinChunkData, [287](#)
- m_compressionMode
 - spinChunkData, [287](#)
- m_compressionRatio
 - spinChunkData, [287](#)
- m_counterValue
 - spinChunkData, [287](#)
- m_cRC
 - spinChunkData, [287](#)
- m_encoderValue
 - spinChunkData, [288](#)
- m_exposureEndLineStatusAll
 - spinChunkData, [288](#)
- m_exposureTime
 - spinChunkData, [288](#)
- m_frameID
 - spinChunkData, [288](#)
- m_gain
 - spinChunkData, [288](#)
- m_height
 - spinChunkData, [288](#)

- m_image
 - spinChunkData, [288](#)
- m_inferenceConfidence
 - spinChunkData, [288](#)
- m_inferenceFrameId
 - spinChunkData, [289](#)
- m_inferenceResult
 - spinChunkData, [289](#)
- m_linePitch
 - spinChunkData, [289](#)
- m_lineStatusAll
 - spinChunkData, [289](#)
- m_offsetX
 - spinChunkData, [289](#)
- m_offsetY
 - spinChunkData, [289](#)
- m_partSelector
 - spinChunkData, [289](#)
- m_pixelDynamicRangeMax
 - spinChunkData, [289](#)
- m_pixelDynamicRangeMin
 - spinChunkData, [290](#)
- m_scan3dAxisMax
 - spinChunkData, [290](#)
- m_scan3dAxisMin
 - spinChunkData, [290](#)
- m_scan3dCoordinateOffset
 - spinChunkData, [290](#)
- m_scan3dCoordinateReferenceValue
 - spinChunkData, [290](#)
- m_scan3dCoordinateScale
 - spinChunkData, [290](#)
- m_scan3dInvalidDataValue
 - spinChunkData, [290](#)
- m_scan3dTransformValue
 - spinChunkData, [290](#)
- m_scanLineSelector
 - spinChunkData, [291](#)
- m_sequencerSetActive
 - spinChunkData, [291](#)
- m_serialDataLength
 - spinChunkData, [291](#)
- m_streamChannelID
 - spinChunkData, [291](#)
- m_timerValue
 - spinChunkData, [291](#)
- m_timestamp
 - spinChunkData, [291](#)
- m_timestampLatchValue
 - spinChunkData, [291](#)
- m_transferBlockID
 - spinChunkData, [291](#)
- m_transferQueueCurrentBlockCount
 - spinChunkData, [292](#)
- m_width
 - spinChunkData, [292](#)
- MACAddress
 - SpinnakerGenApiDefsC.h, [511](#)
- major
 - spinLibraryVersion, [296](#)
- MaxDeviceResetTime
 - quickSpin, [240](#)
- minor
 - spinLibraryVersion, [296](#)
- NA
 - SpinnakerGenApiDefsC.h, [507](#)
- NI
 - SpinnakerGenApiDefsC.h, [507](#)
- No
 - SpinnakerGenApiDefsC.h, [513](#)
- NoCache
 - SpinnakerGenApiDefsC.h, [507](#)
- Node Access, [148](#)
- Node Map Access, [148](#)
- noIncrement
 - SpinnakerGenApiDefsC.h, [508](#)
- None
 - SpinnakerGenApiDefsC.h, [512](#)
- NUM_ACQUISITIONMODE
 - Camera Enumerations, [56](#)
- NUM_ACQUISITIONSTATUSSELECTOR
 - Camera Enumerations, [57](#)
- NUM_ACTIONUNCONDITIONALMODE
 - Camera Enumerations, [57](#)
- NUM_ADCCBITDEPTH
 - Camera Enumerations, [57](#)
- NUM_AUTOALGORITHMSELECTOR
 - Camera Enumerations, [58](#)
- NUM_AUTOEXPOSURECONTROLPRIORITY
 - Camera Enumerations, [58](#)
- NUM_AUTOEXPOSURELIGHTINGMODE
 - Camera Enumerations, [58](#)
- NUM_AUTOEXPOSUREMETERINGMODE
 - Camera Enumerations, [59](#)
- NUM_AUTOEXPOSURETARGETGREYVALUEAUTO
 - Camera Enumerations, [59](#)
- NUM_BALANCERATIOSELECTOR
 - Camera Enumerations, [60](#)
- NUM_BALANCEWHITEAUTO
 - Camera Enumerations, [60](#)
- NUM_BALANCEWHITEAUTOPROFILE
 - Camera Enumerations, [60](#)
- NUM_BINNINGHORIZONTALMODE
 - Camera Enumerations, [61](#)
- NUM_BINNINGSELECTOR
 - Camera Enumerations, [61](#)
- NUM_BINNINGVERTICALMODE
 - Camera Enumerations, [61](#)
- NUM_BLACKLEVELAUTO
 - Camera Enumerations, [62](#)
- NUM_BLACKLEVELAUTOBALANCE
 - Camera Enumerations, [62](#)
- NUM_BLACKLEVELSELECTOR
 - Camera Enumerations, [62](#)
- NUM_CHUNKBLACKLEVELSELECTOR
 - Camera Enumerations, [63](#)

- NUM_CHUNKCOUNTERSELECTOR
 - Camera Enumerations, [63](#)
- NUM_CHUNKENCODERSELECTOR
 - Camera Enumerations, [63](#)
- NUM_CHUNKENCODERSTATUS
 - Camera Enumerations, [63](#)
- NUM_CHUNKEXPOSURETIMESELECTOR
 - Camera Enumerations, [64](#)
- NUM_CHUNKGAINSELECTOR
 - Camera Enumerations, [64](#)
- NUM_CHUNKIMAGECOMPONENT
 - Camera Enumerations, [65](#)
- NUM_CHUNKPIXELFORMAT
 - Camera Enumerations, [65](#)
- NUM_CHUNKREGIONID
 - Camera Enumerations, [65](#)
- NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR
 - Camera Enumerations, [66](#)
- NUM_CHUNKSCAN3DCOORDINATESELECTOR
 - Camera Enumerations, [66](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEM
 - Camera Enumerations, [66](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCESELECTOR
 - Camera Enumerations, [67](#)
- NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR
 - Camera Enumerations, [67](#)
- NUM_CHUNKSCAN3DDISTANCEUNIT
 - Camera Enumerations, [67](#)
- NUM_CHUNKSCAN3DOUTPUTMODE
 - Camera Enumerations, [68](#)
- NUM_CHUNKSELECTOR
 - Camera Enumerations, [69](#)
- NUM_CHUNKSOURCEID
 - Camera Enumerations, [69](#)
- NUM_CHUNKTIMERSELECTOR
 - Camera Enumerations, [70](#)
- NUM_CHUNKTRANSFERSTREAMID
 - Camera Enumerations, [70](#)
- NUM_CLCONFIGURATION
 - Camera Enumerations, [70](#)
- NUM_CLTIMESLOTSCOUNT
 - Camera Enumerations, [71](#)
- NUM_COLORTRANSFORMATIONSELECTOR
 - Camera Enumerations, [71](#)
- NUM_COLORTRANSFORMATIONVALUESELECTOR
 - Camera Enumerations, [71](#)
- NUM_COMPRESSIONSATURATIONPRIORITY
 - Camera Enumerations, [72](#)
- NUM_COUNTEREVENTACTIVATION
 - Camera Enumerations, [72](#)
- NUM_COUNTEREVENTSOURCE
 - Camera Enumerations, [73](#)
- NUM_COUNTERRESETACTIVATION
 - Camera Enumerations, [73](#)
- NUM_COUNTERRESETSOURCE
 - Camera Enumerations, [74](#)
- NUM_COUNTERSELECTOR
 - Camera Enumerations, [74](#)
- NUM_COUNTERSTATUS
 - Camera Enumerations, [74](#)
- NUM_COUNTERTRIGGERACTIVATION
 - Camera Enumerations, [75](#)
- NUM_COUNTERTRIGGERSOURCE
 - Camera Enumerations, [75](#)
- NUM_CXPCONNECTIONTESTMODE
 - Camera Enumerations, [76](#)
- NUM_CXPLINKCONFIGURATION
 - Camera Enumerations, [77](#)
- NUM_CXPLINKCONFIGURATIONPREFERRED
 - Camera Enumerations, [78](#)
- NUM_CXPLINKCONFIGURATIONSTATUS
 - Camera Enumerations, [79](#)
- NUM_CXPPOCXPSTATUS
 - Camera Enumerations, [79](#)
- NUM_DECIMATIONHORIZONTALMODE
 - Camera Enumerations, [79](#)
- NUM_DECIMATIONSELECTOR
 - Camera Enumerations, [80](#)
- NUM_DECIMATIONVERTICALMODE
 - Camera Enumerations, [80](#)
- NUM_DEFECTCORRECTIONMODE
 - Camera Enumerations, [80](#)
- NUM_DEINTERLACING
 - Camera Enumerations, [81](#)
- NUM_DEVICECHARACTERSET
 - Camera Enumerations, [81](#)
- NUM_DEVICECLOCKSELECTOR
 - Camera Enumerations, [81](#)
- NUM_DEVICECONNECTIONSTATUS
 - Camera Enumerations, [81](#)
- NUM_DEVICEINDICATORMODE
 - Camera Enumerations, [82](#)
- NUM_DEVICELINKHEARTBEATMODE
 - Camera Enumerations, [82](#)
- NUM_DEVICELINKTHROUGHPUTLIMITMODE
 - Camera Enumerations, [82](#)
- NUM_DEVICEPOWERSUPPLYSELECTOR
 - Camera Enumerations, [83](#)
- NUM_DEVICEREGISTERSENDIANNESS
 - Camera Enumerations, [83](#)
- NUM_DEVICESCANTYPE
 - Camera Enumerations, [83](#)
- NUM_DEVICESERIALPORTBAUDRATE
 - Camera Enumerations, [84](#)
- NUM_DEVICESERIALPORTSELECTOR
 - Camera Enumerations, [84](#)
- NUM_DEVICESTREAMCHANNELENDIANNESS
 - Camera Enumerations, [84](#)
- NUM_DEVICESTREAMCHANNELTYPE
 - Camera Enumerations, [85](#)
- NUM_DEVICETAPGEOMETRY
 - Camera Enumerations, [86](#)
- NUM_DEVICETEMPERATURESELECTOR
 - Camera Enumerations, [86](#)
- NUM_DEVICETLTYPE
 - Camera Enumerations, [86](#)

- NUM_DEVICETYPE
 - Camera Enumerations, [87](#)
- NUM_ENCODERMODE
 - Camera Enumerations, [87](#)
- NUM_ENCODEROUTPUTMODE
 - Camera Enumerations, [88](#)
- NUM_ENCODERRESETACTIVATION
 - Camera Enumerations, [88](#)
- NUM_ENCODERRESETSOURCE
 - Camera Enumerations, [89](#)
- NUM_ENCODERSELECTOR
 - Camera Enumerations, [89](#)
- NUM_ENCODERSOURCEA
 - Camera Enumerations, [90](#)
- NUM_ENCODERSOURCEB
 - Camera Enumerations, [90](#)
- NUM_ENCODERSTATUS
 - Camera Enumerations, [90](#)
- NUM_EVENTNOTIFICATION
 - Camera Enumerations, [91](#)
- NUM_EVENTSELECTOR
 - Camera Enumerations, [91](#)
- NUM_EXPOSUREACTIVEMODE
 - Camera Enumerations, [91](#)
- NUM_EXPOSUREAUTO
 - Camera Enumerations, [92](#)
- NUM_EXPOSUREMODE
 - Camera Enumerations, [92](#)
- NUM_EXPOSURETIMEMODE
 - Camera Enumerations, [92](#)
- NUM_EXPOSURETIMESELECTOR
 - Camera Enumerations, [93](#)
- NUM_FILEOPENMODE
 - Camera Enumerations, [93](#)
- NUM_FILEOPERATIONSELECTOR
 - Camera Enumerations, [94](#)
- NUM_FILEOPERATIONSTATUS
 - Camera Enumerations, [94](#)
- NUM_FILESELECTOR
 - Camera Enumerations, [94](#)
- NUM_GAINAUTO
 - Camera Enumerations, [96](#)
- NUM_GAINAUTOBALANCE
 - Camera Enumerations, [96](#)
- NUM_GAINSELECTOR
 - Camera Enumerations, [96](#)
- NUM_GEVCPP
 - Camera Enumerations, [97](#)
- NUM_GEVCURRENTPHYSICALLINKCONFIGURATION
 - Camera Enumerations, [97](#)
- NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR
 - Camera Enumerations, [97](#)
- NUM_GEVGVSPEXTENDEDIDMODE
 - Camera Enumerations, [98](#)
- NUM_GEVIDEE1588CLOCKACCURACY
 - Camera Enumerations, [98](#)
- NUM_GEVIDEE1588MODE
 - Camera Enumerations, [98](#)
- NUM_GEVIDEE1588STATUS
 - Camera Enumerations, [99](#)
- NUM_GEVIPCONFIGURATIONSTATUS
 - Camera Enumerations, [99](#)
- NUM_GEVPHYSCALLINKCONFIGURATION
 - Camera Enumerations, [99](#)
- NUM_GEVSUPPORTEDOPTIONSELECTOR
 - Camera Enumerations, [100](#)
- NUM_IMAGECOMPONENTSELECTOR
 - Camera Enumerations, [101](#)
- NUM_IMAGECOMPRESSIONJPEGFORMATOPTION
 - Camera Enumerations, [101](#)
- NUM_IMAGECOMPRESSIONMODE
 - Camera Enumerations, [102](#)
- NUM_IMAGECOMPRESSIONRATEOPTION
 - Camera Enumerations, [102](#)
- NUM_LINEFORMAT
 - Camera Enumerations, [102](#)
- NUM_LINEINPUTFILTERSELECTOR
 - Camera Enumerations, [103](#)
- NUM_LINEMODE
 - Camera Enumerations, [103](#)
- NUM_LINESELECTOR
 - Camera Enumerations, [103](#)
- NUM_LINESOURCE
 - Camera Enumerations, [104](#)
- NUM_LOGICBLOCKLUTINPUTACTIVATION
 - Camera Enumerations, [104](#)
- NUM_LOGICBLOCKLUTINPUTSELECTOR
 - Camera Enumerations, [105](#)
- NUM_LOGICBLOCKLUTINPUTSOURCE
 - Camera Enumerations, [105](#)
- NUM_LOGICBLOCKLUTSELECTOR
 - Camera Enumerations, [106](#)
- NUM_LOGICBLOCKSELECTOR
 - Camera Enumerations, [106](#)
- NUM_LUTSELECTOR
 - Camera Enumerations, [106](#)
- NUM_PIXELCOLORFILTER
 - Camera Enumerations, [107](#)
- NUM_PIXELFORMAT
 - Camera Enumerations, [112](#)
- NUM_PIXELFORMATINFOSELECTOR
 - Camera Enumerations, [118](#)
- NUM_PIXELSIZE
 - Camera Enumerations, [119](#)
- NUM_REGIONDESTINATION
 - Camera Enumerations, [119](#)
- NUM_REGIONMODE
 - Camera Enumerations, [119](#)
- NUM_REGIONSELECTOR
 - Camera Enumerations, [120](#)
- NUM_RGBTRANSFORMLIGHTSOURCE
 - Camera Enumerations, [120](#)
- NUM_SCAN3DCOORDINATEREFERENCESELECTOR
 - Camera Enumerations, [121](#)
- NUM_SCAN3DCOORDINATESELECTOR
 - Camera Enumerations, [121](#)

- NUM_SCAN3DCOORDINATESYSTEM
 - Camera Enumerations, [121](#)
- NUM_SCAN3DCOORDINATESYSTEMREFERENCE
 - Camera Enumerations, [122](#)
- NUM_SCAN3DCOORDINATETRANSFORMSELECTOR
 - Camera Enumerations, [122](#)
- NUM_SCAN3DDISTANCEUNIT
 - Camera Enumerations, [122](#)
- NUM_SCAN3DOUTPUTMODE
 - Camera Enumerations, [124](#)
- NUM_SENSORDIGITIZATIONTAPS
 - Camera Enumerations, [125](#)
- NUM_SENSORSHUTTERMODE
 - Camera Enumerations, [125](#)
- NUM_SENSORTAPS
 - Camera Enumerations, [125](#)
- NUM_SEQUENCERCONFIGURATIONMODE
 - Camera Enumerations, [126](#)
- NUM_SEQUENCERCONFIGURATIONVALID
 - Camera Enumerations, [126](#)
- NUM_SEQUENCERMODE
 - Camera Enumerations, [126](#)
- NUM_SEQUENCERSETVALID
 - Camera Enumerations, [127](#)
- NUM_SEQUENCERTRIGGERACTIVATION
 - Camera Enumerations, [127](#)
- NUM_SEQUENCERTRIGGERSOURCE
 - Camera Enumerations, [127](#)
- NUM_SERIALPORTBAUDRATE
 - Camera Enumerations, [128](#)
- NUM_SERIALPORTPARITY
 - Camera Enumerations, [128](#)
- NUM_SERIALPORTSELECTOR
 - Camera Enumerations, [128](#)
- NUM_SERIALPORTSOURCE
 - Camera Enumerations, [129](#)
- NUM_SERIALPORTSTOPBITS
 - Camera Enumerations, [129](#)
- NUM_SOFTWARESIGNALSELECTOR
 - Camera Enumerations, [129](#)
- NUM_SOURCESELECTOR
 - Camera Enumerations, [130](#)
- NUM_TESTPATTERN
 - Camera Enumerations, [130](#)
- NUM_TESTPATTERNGENERATORSELECTOR
 - Camera Enumerations, [130](#)
- NUM_TIMERSELECTOR
 - Camera Enumerations, [131](#)
- NUM_TIMERSTATUS
 - Camera Enumerations, [131](#)
- NUM_TIMERTRIGGERACTIVATION
 - Camera Enumerations, [131](#)
- NUM_TIMERTRIGGERSOURCE
 - Camera Enumerations, [133](#)
- NUM_TRANSFERCOMPONENTSELECTOR
 - Camera Enumerations, [133](#)
- NUM_TRANSFERCONTROLMODE
 - Camera Enumerations, [134](#)
- NUM_TRANSFEROPERATIONMODE
 - Camera Enumerations, [134](#)
- NUM_TRANSFERQUEUEMODE
 - Camera Enumerations, [134](#)
- NUM_TRANSFERSELECTOR
 - Camera Enumerations, [134](#)
- NUM_TRANSFERSTATUSSELECTOR
 - Camera Enumerations, [135](#)
- NUM_TRANSFERTRIGGERACTIVATION
 - Camera Enumerations, [135](#)
- NUM_TRANSFERTRIGGERMODE
 - Camera Enumerations, [136](#)
- NUM_TRANSFERTRIGGERSELECTOR
 - Camera Enumerations, [136](#)
- NUM_TRANSFERTRIGGERSOURCE
 - Camera Enumerations, [137](#)
- NUM_TRIGGERACTIVATION
 - Camera Enumerations, [137](#)
- NUM_TRIGGERMODE
 - Camera Enumerations, [138](#)
- NUM_TRIGGEROVERLAP
 - Camera Enumerations, [138](#)
- NUM_TRIGGERSELECTOR
 - Camera Enumerations, [138](#)
- NUM_TRIGGERSOURCE
 - Camera Enumerations, [139](#)
- NUM_USEROUTPUTSELECTOR
 - Camera Enumerations, [139](#)
- NUM_USERSETDEFAULT
 - Camera Enumerations, [140](#)
- NUM_USERSETSELECTOR
 - Camera Enumerations, [140](#)
- NUM_WHITECLIPSELECTOR
 - Camera Enumerations, [140](#)
- NUMDEVICEACCESSSTATUS
 - Transport Layer Enumerations, [153](#)
- NUMDEVICECURRENTSPEED
 - Transport Layer Enumerations, [154](#)
- NUMDEVICEENDIANESSMECHANISM
 - Transport Layer Enumerations, [154](#)
- NUMDEVICETYPE
 - Transport Layer Enumerations, [154](#)
- NUMFLIRFILTERDRIVERSTATUS
 - Transport Layer Enumerations, [155](#)
- NUMGENICAMXMLLOCATION
 - Transport Layer Enumerations, [155](#)
- NUMGEVCCP
 - Transport Layer Enumerations, [155](#)
- NUMGUIXMLLOCATION
 - Transport Layer Enumerations, [156](#)
- NUMINTERFACETYPE
 - Transport Layer Enumerations, [156](#)
- NUMPOESTATUS
 - Transport Layer Enumerations, [156](#)
- NUMSTREAMBUFFERCOUNTMODE
 - Transport Layer Enumerations, [157](#)
- NUMSTREAMBUFFERHANDLINGMODE
 - Transport Layer Enumerations, [157](#)

- NUMSTREAMMODE
 - Transport Layer Enumerations, [158](#)
- NUMSTREAMTYPE
 - Transport Layer Enumerations, [158](#)
- NUMTELEDYNEGIGEVISIONFILTERDRIVERSTATUS
 - Transport Layer Enumerations, [158](#)
- NUMTLTYPE
 - Transport Layer Enumerations, [159](#)
- OffsetX
 - quickSpin, [241](#)
- OffsetY
 - quickSpin, [241](#)
- PacketResendRequestCount
 - quickSpin, [241](#)
- PayloadSize
 - quickSpin, [241](#)
- pblsStreaming
 - SpinnakerC.h, [446](#)
- PixelColorFilter
 - quickSpin, [241](#)
- PixelColorFilter_BayerBG
 - Camera Enumerations, [107](#)
- PixelColorFilter_BayerGB
 - Camera Enumerations, [107](#)
- PixelColorFilter_BayerGR
 - Camera Enumerations, [107](#)
- PixelColorFilter_BayerRG
 - Camera Enumerations, [106](#)
- PixelColorFilter_None
 - Camera Enumerations, [106](#)
- PixelDynamicRangeMax
 - quickSpin, [241](#)
- PixelDynamicRangeMin
 - quickSpin, [241](#)
- PixelFormat
 - quickSpin, [241](#)
- PixelFormat_B10
 - Camera Enumerations, [109](#)
- PixelFormat_B12
 - Camera Enumerations, [109](#)
- PixelFormat_B12_Jpeg
 - Camera Enumerations, [112](#)
- PixelFormat_B16
 - Camera Enumerations, [109](#)
- PixelFormat_B8
 - Camera Enumerations, [109](#)
- PixelFormat_BayerBG10
 - Camera Enumerations, [108](#)
- PixelFormat_BayerBG10p
 - Camera Enumerations, [108](#)
- PixelFormat_BayerBG10Packed
 - Camera Enumerations, [108](#)
- PixelFormat_BayerBG12
 - Camera Enumerations, [108](#)
- PixelFormat_BayerBG12p
 - Camera Enumerations, [107](#)
- PixelFormat_BayerBG12Packed
 - Camera Enumerations, [107](#)
- PixelFormat_BayerBG16
 - Camera Enumerations, [107](#)
- PixelFormat_BayerBG8
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGB10
 - Camera Enumerations, [108](#)
- PixelFormat_BayerGB10p
 - Camera Enumerations, [108](#)
- PixelFormat_BayerGB10Packed
 - Camera Enumerations, [108](#)
- PixelFormat_BayerGB12
 - Camera Enumerations, [108](#)
- PixelFormat_BayerGB12p
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGB12Packed
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGB16
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGB8
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGR10
 - Camera Enumerations, [108](#)
- PixelFormat_BayerGR10p
 - Camera Enumerations, [108](#)
- PixelFormat_BayerGR10Packed
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGR12
 - Camera Enumerations, [108](#)
- PixelFormat_BayerGR12p
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGR12Packed
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGR16
 - Camera Enumerations, [107](#)
- PixelFormat_BayerGR8
 - Camera Enumerations, [107](#)
- PixelFormat_BayerRG10
 - Camera Enumerations, [108](#)
- PixelFormat_BayerRG10p
 - Camera Enumerations, [108](#)
- PixelFormat_BayerRG10Packed
 - Camera Enumerations, [107](#)
- PixelFormat_BayerRG12
 - Camera Enumerations, [108](#)
- PixelFormat_BayerRG12p
 - Camera Enumerations, [107](#)
- PixelFormat_BayerRG12Packed
 - Camera Enumerations, [107](#)
- PixelFormat_BayerRG16
 - Camera Enumerations, [107](#)
- PixelFormat_BayerRG8
 - Camera Enumerations, [107](#)
- PixelFormat_BayerRGPolarized10p
 - Camera Enumerations, [112](#)
- PixelFormat_BayerRGPolarized12p
 - Camera Enumerations, [112](#)
- PixelFormat_BayerRGPolarized16
 - Camera Enumerations, [107](#)

Camera Enumerations, [112](#)
PixelFormat_BayerRGPolarized8
Camera Enumerations, [112](#)
PixelFormat_BGR10
Camera Enumerations, [109](#)
PixelFormat_BGR10p
Camera Enumerations, [109](#)
PixelFormat_BGR12
Camera Enumerations, [109](#)
PixelFormat_BGR12p
Camera Enumerations, [109](#)
PixelFormat_BGR14
Camera Enumerations, [109](#)
PixelFormat_BGR16
Camera Enumerations, [109](#)
PixelFormat_BGR565p
Camera Enumerations, [109](#)
PixelFormat_BGR8
Camera Enumerations, [107](#)
PixelFormat_BGRa10
Camera Enumerations, [109](#)
PixelFormat_BGRa10p
Camera Enumerations, [109](#)
PixelFormat_BGRa12
Camera Enumerations, [109](#)
PixelFormat_BGRa12p
Camera Enumerations, [109](#)
PixelFormat_BGRa14
Camera Enumerations, [109](#)
PixelFormat_BGRa16
Camera Enumerations, [109](#)
PixelFormat_BGRa8
Camera Enumerations, [107](#)
PixelFormat_BiColorBGRG10
Camera Enumerations, [110](#)
PixelFormat_BiColorBGRG10p
Camera Enumerations, [110](#)
PixelFormat_BiColorBGRG12
Camera Enumerations, [110](#)
PixelFormat_BiColorBGRG12p
Camera Enumerations, [110](#)
PixelFormat_BiColorBGRG8
Camera Enumerations, [110](#)
PixelFormat_BiColorRGBG10
Camera Enumerations, [110](#)
PixelFormat_BiColorRGBG10p
Camera Enumerations, [110](#)
PixelFormat_BiColorRGBG12
Camera Enumerations, [110](#)
PixelFormat_BiColorRGBG12p
Camera Enumerations, [110](#)
PixelFormat_BiColorRGBG8
Camera Enumerations, [110](#)
PixelFormat_Confidence1
Camera Enumerations, [110](#)
PixelFormat_Confidence16
Camera Enumerations, [110](#)
PixelFormat_Confidence1p

Camera Enumerations, [110](#)
PixelFormat_Confidence32f
Camera Enumerations, [110](#)
PixelFormat_Confidence8
Camera Enumerations, [110](#)
PixelFormat_Coord3D_A10p
Camera Enumerations, [110](#)
PixelFormat_Coord3D_A12p
Camera Enumerations, [110](#)
PixelFormat_Coord3D_A16
Camera Enumerations, [110](#)
PixelFormat_Coord3D_A32f
Camera Enumerations, [110](#)
PixelFormat_Coord3D_A8
Camera Enumerations, [110](#)
PixelFormat_Coord3D_ABC10p
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC10p_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC12p
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC12p_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC16
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC16_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC32f
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC32f_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC8
Camera Enumerations, [109](#)
PixelFormat_Coord3D_ABC8_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC10p
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC10p_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC12p
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC12p_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC16
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC16_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC32f
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC32f_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC8
Camera Enumerations, [109](#)
PixelFormat_Coord3D_AC8_Planar
Camera Enumerations, [109](#)
PixelFormat_Coord3D_B10p
Camera Enumerations, [110](#)
PixelFormat_Coord3D_B12p

- Camera Enumerations, [110](#)
- PixelFormat_Coord3D_B16
 - Camera Enumerations, [110](#)
- PixelFormat_Coord3D_B32f
 - Camera Enumerations, [110](#)
- PixelFormat_Coord3D_B8
 - Camera Enumerations, [110](#)
- PixelFormat_Coord3D_C10p
 - Camera Enumerations, [110](#)
- PixelFormat_Coord3D_C12p
 - Camera Enumerations, [110](#)
- PixelFormat_Coord3D_C16
 - Camera Enumerations, [110](#)
- PixelFormat_Coord3D_C32f
 - Camera Enumerations, [110](#)
- PixelFormat_Coord3D_C8
 - Camera Enumerations, [110](#)
- PixelFormat_G10
 - Camera Enumerations, [109](#)
- PixelFormat_G12
 - Camera Enumerations, [109](#)
- PixelFormat_G16
 - Camera Enumerations, [109](#)
- PixelFormat_G8
 - Camera Enumerations, [109](#)
- PixelFormat_GB12
 - Camera Enumerations, [112](#)
- PixelFormat_GB12_Jpeg
 - Camera Enumerations, [112](#)
- PixelFormat_GR12
 - Camera Enumerations, [112](#)
- PixelFormat_GR12_Jpeg
 - Camera Enumerations, [112](#)
- PixelFormat_JPEGColor8
 - Camera Enumerations, [112](#)
- PixelFormat_JPEGMono8
 - Camera Enumerations, [112](#)
- PixelFormat_LLCBayerRG8
 - Camera Enumerations, [112](#)
- PixelFormat_LLCMono8
 - Camera Enumerations, [112](#)
- PixelFormat_Mono10
 - Camera Enumerations, [108](#)
- PixelFormat_Mono10p
 - Camera Enumerations, [108](#)
- PixelFormat_Mono10Packed
 - Camera Enumerations, [107](#)
- PixelFormat_Mono12
 - Camera Enumerations, [108](#)
- PixelFormat_Mono12p
 - Camera Enumerations, [107](#)
- PixelFormat_Mono12Packed
 - Camera Enumerations, [107](#)
- PixelFormat_Mono14
 - Camera Enumerations, [108](#)
- PixelFormat_Mono16
 - Camera Enumerations, [107](#)
- PixelFormat_Mono16s
 - Camera Enumerations, [108](#)
- PixelFormat_Mono1p
 - Camera Enumerations, [108](#)
- PixelFormat_Mono2p
 - Camera Enumerations, [108](#)
- PixelFormat_Mono32f
 - Camera Enumerations, [108](#)
- PixelFormat_Mono4p
 - Camera Enumerations, [108](#)
- PixelFormat_Mono8
 - Camera Enumerations, [107](#)
- PixelFormat_Mono8s
 - Camera Enumerations, [108](#)
- PixelFormat_Polarized10p
 - Camera Enumerations, [112](#)
- PixelFormat_Polarized12p
 - Camera Enumerations, [112](#)
- PixelFormat_Polarized16
 - Camera Enumerations, [112](#)
- PixelFormat_Polarized8
 - Camera Enumerations, [112](#)
- PixelFormat_R10
 - Camera Enumerations, [109](#)
- PixelFormat_R12
 - Camera Enumerations, [109](#)
- PixelFormat_R12_Jpeg
 - Camera Enumerations, [112](#)
- PixelFormat_R16
 - Camera Enumerations, [109](#)
- PixelFormat_R8
 - Camera Enumerations, [109](#)
- PixelFormat_Raw16
 - Camera Enumerations, [112](#)
- PixelFormat_Raw8
 - Camera Enumerations, [112](#)
- PixelFormat_RGB10
 - Camera Enumerations, [108](#)
- PixelFormat_RGB10_Planar
 - Camera Enumerations, [108](#)
- PixelFormat_RGB10p
 - Camera Enumerations, [108](#)
- PixelFormat_RGB10p32
 - Camera Enumerations, [108](#)
- PixelFormat_RGB12
 - Camera Enumerations, [108](#)
- PixelFormat_RGB12_Planar
 - Camera Enumerations, [108](#)
- PixelFormat_RGB12p
 - Camera Enumerations, [108](#)
- PixelFormat_RGB14
 - Camera Enumerations, [108](#)
- PixelFormat_RGB16
 - Camera Enumerations, [108](#)
- PixelFormat_RGB16_Planar
 - Camera Enumerations, [108](#)
- PixelFormat_RGB16s
 - Camera Enumerations, [108](#)
- PixelFormat_RGB32f
 - Camera Enumerations, [108](#)

- Camera Enumerations, [108](#)
- PixelFormat_RGB565p
 - Camera Enumerations, [109](#)
- PixelFormat_RGB8
 - Camera Enumerations, [108](#)
- PixelFormat_RGB8_Planar
 - Camera Enumerations, [108](#)
- PixelFormat_RGB8Packed
 - Camera Enumerations, [107](#)
- PixelFormat_RGBa10
 - Camera Enumerations, [108](#)
- PixelFormat_RGBa10p
 - Camera Enumerations, [108](#)
- PixelFormat_RGBa12
 - Camera Enumerations, [108](#)
- PixelFormat_RGBa12p
 - Camera Enumerations, [108](#)
- PixelFormat_RGBa14
 - Camera Enumerations, [108](#)
- PixelFormat_RGBa16
 - Camera Enumerations, [108](#)
- PixelFormat_RGBa32f
 - Camera Enumerations, [109](#)
- PixelFormat_RGBa8
 - Camera Enumerations, [108](#)
- PixelFormat_SCF1WBWG10
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WBWG10p
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WBWG12
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WBWG12p
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WBWG14
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WBWG16
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WBWG8
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWB10
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWB10p
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWB12
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWB12p
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWB14
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWB16
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWB8
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWR10
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWR10p
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WGWR12
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WGWR12p
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WGWR14
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WGWR16
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WGWR8
 - Camera Enumerations, [110](#)
- PixelFormat_SCF1WRWG10
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WRWG10p
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WRWG12
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WRWG12p
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WRWG14
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WRWG16
 - Camera Enumerations, [111](#)
- PixelFormat_SCF1WRWG8
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr10_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr10p_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr12_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr12p_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr411_8
 - Camera Enumerations, [107](#)
- PixelFormat_YCbCr411_8_CbYYCrYY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_10
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_10_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_10p
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_10p_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_12
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_12_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_12p
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_12p_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr422_8
 - Camera Enumerations, [107](#)
- PixelFormat_YCbCr422_8_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_10_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_10p_CbYCr

- Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_12_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_12p_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_411_8_CbYYCrYY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_10
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_10_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_10p
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_10p_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_12
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_12_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_12p
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_12p_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_8
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_422_8_CbYCrY
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr601_8_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr709_10_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr709_10p_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr709_12_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr709_12p_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr709_411_8_CbYYCrYY
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_10
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_10_CbYCrY
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_10p
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_10p_CbYCrY
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_12
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_12_CbYCrY
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_12p
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_12p_CbYCrY
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_8
 - Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_422_8_CbYCrY
 - Camera Enumerations, [112](#)
- Camera Enumerations, [112](#)
- PixelFormat_YCbCr709_8_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YCbCr8
 - Camera Enumerations, [107](#)
- PixelFormat_YCbCr8_CbYCr
 - Camera Enumerations, [111](#)
- PixelFormat_YUV411_8_UYYVYY
 - Camera Enumerations, [112](#)
- PixelFormat_YUV411Packed
 - Camera Enumerations, [107](#)
- PixelFormat_YUV422_8
 - Camera Enumerations, [112](#)
- PixelFormat_YUV422_8_UYVY
 - Camera Enumerations, [112](#)
- PixelFormat_YUV422Packed
 - Camera Enumerations, [107](#)
- PixelFormat_YUV444Packed
 - Camera Enumerations, [107](#)
- PixelFormat_YUV8_UYV
 - Camera Enumerations, [112](#)
- PixelFormatInfoD
 - quickSpin, [242](#)
- PixelFormatInfoSelector
 - quickSpin, [242](#)
- PixelFormatInfoSelector_B10
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_B12
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_B16
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_B8
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_BayerBG10
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerBG10p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerBG12
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerBG12p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerBG16
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerBG8
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerGB10
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerGB10p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerGB12
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerGB12p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerGB16
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerGB8
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_BayerGR10
 - Camera Enumerations, [113](#)

Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerGR10p
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerGR12
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerGR12p
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerGR16
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerGR8
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerRG10
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerRG10p
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerRG12
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerRG12p
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerRG16
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerRG8
Camera Enumerations, [113](#)
PixelFormatInfoSelector_BayerRGPolarized10p
Camera Enumerations, [118](#)
PixelFormatInfoSelector_BayerRGPolarized12p
Camera Enumerations, [118](#)
PixelFormatInfoSelector_BayerRGPolarized16
Camera Enumerations, [118](#)
PixelFormatInfoSelector_BayerRGPolarized8
Camera Enumerations, [118](#)
PixelFormatInfoSelector_BGR10
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGR10p
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGR12
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGR12p
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGR14
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGR16
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGR565p
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGR8
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGRa10
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGRa10p
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGRa12
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGRa12p
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGRa14
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGRa16

Camera Enumerations, [114](#)
PixelFormatInfoSelector_BGRa8
Camera Enumerations, [114](#)
PixelFormatInfoSelector_BiColorBGRG10
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorBGRG10p
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorBGRG12
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorBGRG12p
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorBGRG8
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorRGBG10
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorRGBG10p
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorRGBG12
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorRGBG12p
Camera Enumerations, [115](#)
PixelFormatInfoSelector_BiColorRGBG8
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Confidence1
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Confidence16
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Confidence1p
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Confidence32f
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Confidence8
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_A10p
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_A12p
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_A16
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_A32f
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_A8
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_ABC10p
Camera Enumerations, [114](#)
PixelFormatInfoSelector_Coord3D_ABC10p_Planar
Camera Enumerations, [114](#)
PixelFormatInfoSelector_Coord3D_ABC12p
Camera Enumerations, [114](#)
PixelFormatInfoSelector_Coord3D_ABC12p_Planar
Camera Enumerations, [114](#)
PixelFormatInfoSelector_Coord3D_ABC16
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_ABC16_Planar
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_ABC32f
Camera Enumerations, [115](#)
PixelFormatInfoSelector_Coord3D_ABC32f_Planar

- Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_ABC8
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_Coord3D_ABC8_Planar
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_Coord3D_AC10p
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC10p_Planar
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC12p
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC12p_Planar
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC16
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC16_Planar
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC32f
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC32f_Planar
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC8
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_AC8_Planar
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_B10p
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_B12p
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_B16
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_B32f
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_B8
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_C10p
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_C12p
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_C16
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_C32f
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_Coord3D_C8
 - Camera Enumerations, [115](#)
- PixelFormatInfoSelector_G10
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_G12
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_G16
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_G8
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_JPEGColor8
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_JPEGMono8
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_LLCBayerRG8
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_LLCMono8
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_Mono10
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono10p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono12
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono12p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono14
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono16
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono16s
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono1p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono2p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono32f
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono4p
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono8
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Mono8s
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_Polarized10p
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_Polarized12p
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_Polarized16
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_Polarized8
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_R10
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_R12
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_R16
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_R8
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_RGB10
 - Camera Enumerations, [113](#)
- PixelFormatInfoSelector_RGB10_Planar
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_RGB10p
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_RGB10p32
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_RGB12
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_RGB12_Planar
 - Camera Enumerations, [114](#)
- PixelFormatInfoSelector_RGB12p
 - Camera Enumerations, [114](#)

Camera Enumerations, [114](#)
PixelFormatInfoSelector_RGB14
Camera Enumerations, [114](#)
PixelFormatInfoSelector_RGB16
Camera Enumerations, [114](#)
PixelFormatInfoSelector_RGB16_Planar
Camera Enumerations, [114](#)
PixelFormatInfoSelector_RGB16s
Camera Enumerations, [114](#)
PixelFormatInfoSelector_RGB32f
Camera Enumerations, [114](#)
PixelFormatInfoSelector_RGB565p
Camera Enumerations, [114](#)
PixelFormatInfoSelector_RGB8
Camera Enumerations, [113](#)
PixelFormatInfoSelector_RGB8_Planar
Camera Enumerations, [113](#)
PixelFormatInfoSelector_RGBa10
Camera Enumerations, [113](#)
PixelFormatInfoSelector_RGBa10p
Camera Enumerations, [113](#)
PixelFormatInfoSelector_RGBa12
Camera Enumerations, [113](#)
PixelFormatInfoSelector_RGBa12p
Camera Enumerations, [113](#)
PixelFormatInfoSelector_RGBa14
Camera Enumerations, [113](#)
PixelFormatInfoSelector_RGBa16
Camera Enumerations, [113](#)
PixelFormatInfoSelector_RGBa32f
Camera Enumerations, [114](#)
PixelFormatInfoSelector_RGBa8
Camera Enumerations, [113](#)
PixelFormatInfoSelector_SCF1WBWG10
Camera Enumerations, [115](#)
PixelFormatInfoSelector_SCF1WBWG10p
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WBWG12
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WBWG12p
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WBWG14
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WBWG16
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WBWG8
Camera Enumerations, [115](#)
PixelFormatInfoSelector_SCF1WGWB10
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWB10p
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWB12
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWB12p
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWB14
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWB16

Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWB8
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWR10
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWR10p
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWR12
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWR12p
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWR14
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWR16
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WGWR8
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WRWG10
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WRWG10p
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WRWG12
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WRWG12p
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WRWG14
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WRWG16
Camera Enumerations, [116](#)
PixelFormatInfoSelector_SCF1WRWG8
Camera Enumerations, [116](#)
PixelFormatInfoSelector_YCbCr10_CbYCr
Camera Enumerations, [116](#)
PixelFormatInfoSelector_YCbCr10p_CbYCr
Camera Enumerations, [116](#)
PixelFormatInfoSelector_YCbCr12_CbYCr
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr12p_CbYCr
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr411_8
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr422_10
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr422_10_CbYCrY
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr422_10p
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr422_12
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr422_12_CbYCrY
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr422_12p
Camera Enumerations, [117](#)
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY

- Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr422_8
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr422_8_CbYCrY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_10_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_10p_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_12_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_12p_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_10
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_10p
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_12
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_12p
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_8
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr601_8_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_10_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_10p_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_12_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_12p_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_422_10
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_422_10p
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_YCbCr709_422_12
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY
 - Camera Enumerations, [118](#)
- Camera Enumerations, [118](#)
- PixelFormatInfoSelector_YCbCr709_422_12p
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_YCbCr709_422_8
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr709_8_CbYCr
 - Camera Enumerations, [117](#)
- PixelFormatInfoSelector_YCbCr8
 - Camera Enumerations, [116](#)
- PixelFormatInfoSelector_YCbCr8_CbYCr
 - Camera Enumerations, [116](#)
- PixelFormatInfoSelector_YUV411_8_UYYVYY
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_YUV422_8
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_YUV422_8_UYVY
 - Camera Enumerations, [118](#)
- PixelFormatInfoSelector_YUV8_UYV
 - Camera Enumerations, [118](#)
- PixelSize
 - quickSpin, [242](#)
- PixelSize_Bpp1
 - Camera Enumerations, [118](#)
- PixelSize_Bpp10
 - Camera Enumerations, [118](#)
- PixelSize_Bpp12
 - Camera Enumerations, [118](#)
- PixelSize_Bpp14
 - Camera Enumerations, [118](#)
- PixelSize_Bpp16
 - Camera Enumerations, [118](#)
- PixelSize_Bpp2
 - Camera Enumerations, [118](#)
- PixelSize_Bpp20
 - Camera Enumerations, [118](#)
- PixelSize_Bpp24
 - Camera Enumerations, [118](#)
- PixelSize_Bpp30
 - Camera Enumerations, [118](#)
- PixelSize_Bpp32
 - Camera Enumerations, [118](#)
- PixelSize_Bpp36
 - Camera Enumerations, [119](#)
- PixelSize_Bpp4
 - Camera Enumerations, [118](#)
- PixelSize_Bpp48
 - Camera Enumerations, [119](#)
- PixelSize_Bpp64
 - Camera Enumerations, [119](#)
- PixelSize_Bpp8
 - Camera Enumerations, [118](#)
- PixelSize_Bpp96
 - Camera Enumerations, [119](#)
- POEStatus

- quickSpinTLInterface, [273](#)
- POEStatus_NotSupported
 - Transport Layer Enumerations, [156](#)
- POEStatus_PowerOff
 - Transport Layer Enumerations, [156](#)
- POEStatus_PowerOn
 - Transport Layer Enumerations, [156](#)
- PortNode
 - SpinnakerGenApiDefsC.h, [511](#)
- PowerSupplyCurrent
 - quickSpin, [242](#)
- PowerSupplyVoltage
 - quickSpin, [242](#)
- progressive
 - spinJPEGOption, [294](#)
- PureNumber
 - SpinnakerGenApiDefsC.h, [511](#)
- quality
 - spinJPEGOption, [294](#)
 - spinJPG2Option, [295](#)
 - spinMJPEGOption, [298](#)
- quickSpin, [162](#)
 - AasRoiEnable, [174](#)
 - AasRoiHeight, [174](#)
 - AasRoiOffsetX, [174](#)
 - AasRoiOffsetY, [174](#)
 - AasRoiWidth, [174](#)
 - AcquisitionAbort, [175](#)
 - AcquisitionArm, [175](#)
 - AcquisitionBurstFrameCount, [175](#)
 - AcquisitionFrameCount, [175](#)
 - AcquisitionFrameRate, [175](#)
 - AcquisitionFrameRateEnable, [175](#)
 - AcquisitionLineRate, [175](#)
 - AcquisitionMode, [175](#)
 - AcquisitionResultingFrameRate, [176](#)
 - AcquisitionStart, [176](#)
 - AcquisitionStatus, [176](#)
 - AcquisitionStatusSelector, [176](#)
 - AcquisitionStop, [176](#)
 - ActionDeviceKey, [176](#)
 - ActionGroupKey, [176](#)
 - ActionGroupMask, [176](#)
 - ActionQueueSize, [177](#)
 - ActionSelector, [177](#)
 - ActionUnconditionalMode, [177](#)
 - AdaptiveCompressionEnable, [177](#)
 - AdcBitDepth, [177](#)
 - aPAUSEMACCtrlFramesReceived, [177](#)
 - aPAUSEMACCtrlFramesTransmitted, [177](#)
 - AutoAlgorithmSelector, [177](#)
 - AutoExposureControlLoopDamping, [178](#)
 - AutoExposureControlPriority, [178](#)
 - AutoExposureEVCompensation, [178](#)
 - AutoExposureExposureTimeLowerLimit, [178](#)
 - AutoExposureExposureTimeUpperLimit, [178](#)
 - AutoExposureGainLowerLimit, [178](#)
 - AutoExposureGainUpperLimit, [178](#)
 - AutoExposureGreyValueLowerLimit, [178](#)
 - AutoExposureGreyValueUpperLimit, [179](#)
 - AutoExposureLightingMode, [179](#)
 - AutoExposureMeteringMode, [179](#)
 - AutoExposureTargetGreyValue, [179](#)
 - AutoExposureTargetGreyValueAuto, [179](#)
 - BalanceRatio, [179](#)
 - BalanceRatioSelector, [179](#)
 - BalanceWhiteAuto, [179](#)
 - BalanceWhiteAutoDamping, [180](#)
 - BalanceWhiteAutoLowerLimit, [180](#)
 - BalanceWhiteAutoProfile, [180](#)
 - BalanceWhiteAutoUpperLimit, [180](#)
 - BinningHorizontal, [180](#)
 - BinningHorizontalMode, [180](#)
 - BinningSelector, [180](#)
 - BinningVertical, [180](#)
 - BinningVerticalMode, [181](#)
 - BlackLevel, [181](#)
 - BlackLevelAuto, [181](#)
 - BlackLevelAutoBalance, [181](#)
 - BlackLevelClampingEnable, [181](#)
 - BlackLevelRaw, [181](#)
 - BlackLevelSelector, [181](#)
 - ChunkBlackLevel, [181](#)
 - ChunkBlackLevelSelector, [182](#)
 - ChunkCompressionMode, [182](#)
 - ChunkCompressionRatio, [182](#)
 - ChunkCounterSelector, [182](#)
 - ChunkCounterValue, [182](#)
 - ChunkCRC, [182](#)
 - ChunkEnable, [182](#)
 - ChunkEncoderSelector, [182](#)
 - ChunkEncoderStatus, [183](#)
 - ChunkEncoderValue, [183](#)
 - ChunkExposureEndLineStatusAll, [183](#)
 - ChunkExposureTime, [183](#)
 - ChunkExposureTimeSelector, [183](#)
 - ChunkFrameID, [183](#)
 - ChunkGain, [183](#)
 - ChunkGainSelector, [183](#)
 - ChunkHeight, [184](#)
 - ChunkImage, [184](#)
 - ChunkImageComponent, [184](#)
 - ChunkInferenceBoundingBoxResult, [184](#)
 - ChunkInferenceConfidence, [184](#)
 - ChunkInferenceFrameId, [184](#)
 - ChunkInferenceResult, [184](#)
 - ChunkLinePitch, [184](#)
 - ChunkLineStatusAll, [185](#)
 - ChunkModeActive, [185](#)
 - ChunkOffsetX, [185](#)
 - ChunkOffsetY, [185](#)
 - ChunkPartSelector, [185](#)
 - ChunkPixelDynamicRangeMax, [185](#)
 - ChunkPixelDynamicRangeMin, [185](#)
 - ChunkPixelFormat, [185](#)
 - ChunkRegionID, [186](#)

- ChunkScan3dAxisMax, 186
- ChunkScan3dAxisMin, 186
- ChunkScan3dCoordinateOffset, 186
- ChunkScan3dCoordinateReferenceSelector, 186
- ChunkScan3dCoordinateReferenceValue, 186
- ChunkScan3dCoordinateScale, 186
- ChunkScan3dCoordinateSelector, 186
- ChunkScan3dCoordinateSystem, 187
- ChunkScan3dCoordinateSystemReference, 187
- ChunkScan3dCoordinateTransformSelector, 187
- ChunkScan3dDistanceUnit, 187
- ChunkScan3dInvalidDataFlag, 187
- ChunkScan3dInvalidDataValue, 187
- ChunkScan3dOutputMode, 187
- ChunkScan3dTransformValue, 187
- ChunkScanLineSelector, 188
- ChunkSelector, 188
- ChunkSequencerSetActive, 188
- ChunkSerialData, 188
- ChunkSerialDataLength, 188
- ChunkSerialReceiveOverflow, 188
- ChunkSourceID, 188
- ChunkStreamChannelID, 188
- ChunkTimerSelector, 189
- ChunkTimerValue, 189
- ChunkTimestamp, 189
- ChunkTimestampLatchValue, 189
- ChunkTransferBlockID, 189
- ChunkTransferQueueCurrentBlockCount, 189
- ChunkTransferStreamID, 189
- ChunkWidth, 189
- CIConfiguration, 190
- CITimeSlotsCount, 190
- ColorTransformationEnable, 190
- ColorTransformationSelector, 190
- ColorTransformationValue, 190
- ColorTransformationValueSelector, 190
- CompressionRatio, 190
- CompressionSaturationPriority, 190
- CounterDelay, 191
- CounterDuration, 191
- CounterEventActivation, 191
- CounterEventSource, 191
- CounterReset, 191
- CounterResetActivation, 191
- CounterResetSource, 191
- CounterSelector, 191
- CounterStatus, 192
- CounterTriggerActivation, 192
- CounterTriggerSource, 192
- CounterValue, 192
- CounterValueAtReset, 192
- CxpConnectionSelector, 192
- CxpConnectionTestErrorCount, 192
- CxpConnectionTestMode, 192
- CxpConnectionTestPacketCount, 193
- CxpLinkConfiguration, 193
- CxpLinkConfigurationPreferred, 193
- CxpLinkConfigurationStatus, 193
- CxpPoCxpAuto, 193
- CxpPoCxpStatus, 193
- CxpPoCxpTripReset, 193
- CxpPoCxpTurnOff, 193
- DecimationHorizontal, 194
- DecimationHorizontalMode, 194
- DecimationSelector, 194
- DecimationVertical, 194
- DecimationVerticalMode, 194
- DefectCorrectionMode, 194
- DefectCorrectStaticEnable, 194
- DefectTableApply, 194
- DefectTableCoordinateX, 195
- DefectTableCoordinateY, 195
- DefectTableFactoryRestore, 195
- DefectTableIndex, 195
- DefectTablePixelCount, 195
- DefectTableSave, 195
- Deinterlacing, 195
- DeviceCharacterSet, 195
- DeviceClockFrequency, 196
- DeviceClockSelector, 196
- DeviceConnectionSelector, 196
- DeviceConnectionSpeed, 196
- DeviceConnectionStatus, 196
- DeviceEventChannelCount, 196
- DeviceFamilyName, 196
- DeviceFeaturePersistenceEnd, 196
- DeviceFeaturePersistenceStart, 197
- DeviceFirmwareVersion, 197
- DeviceGenCPVersionMajor, 197
- DeviceGenCPVersionMinor, 197
- DeviceID, 197
- DeviceIndicatorMode, 197
- DeviceLinkBandwidthReserve, 197
- DeviceLinkCommandTimeout, 197
- DeviceLinkConnectionCount, 198
- DeviceLinkCurrentThroughput, 198
- DeviceLinkHeartbeatMode, 198
- DeviceLinkHeartbeatTimeout, 198
- DeviceLinkSelector, 198
- DeviceLinkSpeed, 198
- DeviceLinkThroughputLimit, 198
- DeviceLinkThroughputLimitMode, 198
- DeviceManifestEntrySelector, 199
- DeviceManifestPrimaryURL, 199
- DeviceManifestSchemaMajorVersion, 199
- DeviceManifestSchemaMinorVersion, 199
- DeviceManifestSecondaryURL, 199
- DeviceManifestXMLMajorVersion, 199
- DeviceManifestXMLMinorVersion, 199
- DeviceManifestXMLSubMinorVersion, 199
- DeviceManufacturerInfo, 200
- DeviceMaxThroughput, 200
- DeviceModelName, 200
- DevicePowerSupplySelector, 200
- DeviceRegistersCheck, 200

- DeviceRegistersEndianness, [200](#)
- DeviceRegistersStreamingEnd, [200](#)
- DeviceRegistersStreamingStart, [200](#)
- DeviceRegistersValid, [201](#)
- DeviceReset, [201](#)
- DeviceScanType, [201](#)
- DeviceSerialNumber, [201](#)
- DeviceSerialPortBaudRate, [201](#)
- DeviceSerialPortSelector, [201](#)
- DeviceSFNCVersionMajor, [201](#)
- DeviceSFNCVersionMinor, [201](#)
- DeviceSFNCVersionSubMinor, [202](#)
- DeviceStreamChannelCount, [202](#)
- DeviceStreamChannelEndianness, [202](#)
- DeviceStreamChannelLink, [202](#)
- DeviceStreamChannelPacketSize, [202](#)
- DeviceStreamChannelSelector, [202](#)
- DeviceStreamChannelType, [202](#)
- DeviceTapGeometry, [202](#)
- DeviceTemperature, [203](#)
- DeviceTemperatureSelector, [203](#)
- DeviceTLType, [203](#)
- DeviceTLVersionMajor, [203](#)
- DeviceTLVersionMinor, [203](#)
- DeviceTLVersionSubMinor, [203](#)
- DeviceType, [203](#)
- DeviceUptime, [203](#)
- DeviceUserID, [204](#)
- DeviceVendorName, [204](#)
- DeviceVersion, [204](#)
- EncoderDivider, [204](#)
- EncoderMode, [204](#)
- EncoderOutputMode, [204](#)
- EncoderReset, [204](#)
- EncoderResetActivation, [204](#)
- EncoderResetSource, [205](#)
- EncoderSelector, [205](#)
- EncoderSourceA, [205](#)
- EncoderSourceB, [205](#)
- EncoderStatus, [205](#)
- EncoderTimeout, [205](#)
- EncoderValue, [205](#)
- EncoderValueAtReset, [205](#)
- EnumerationCount, [206](#)
- EventAcquisitionEnd, [206](#)
- EventAcquisitionEndFrameID, [206](#)
- EventAcquisitionEndTimestamp, [206](#)
- EventAcquisitionError, [206](#)
- EventAcquisitionErrorFrameID, [206](#)
- EventAcquisitionErrorTimestamp, [206](#)
- EventAcquisitionStart, [206](#)
- EventAcquisitionStartFrameID, [207](#)
- EventAcquisitionStartTimestamp, [207](#)
- EventAcquisitionTransferEnd, [207](#)
- EventAcquisitionTransferEndFrameID, [207](#)
- EventAcquisitionTransferEndTimestamp, [207](#)
- EventAcquisitionTransferStart, [207](#)
- EventAcquisitionTransferStartFrameID, [207](#)
- EventAcquisitionTransferStartTimestamp, [207](#)
- EventAcquisitionTrigger, [208](#)
- EventAcquisitionTriggerFrameID, [208](#)
- EventAcquisitionTriggerTimestamp, [208](#)
- EventActionLate, [208](#)
- EventActionLateFrameID, [208](#)
- EventActionLateTimestamp, [208](#)
- EventCounter0End, [208](#)
- EventCounter0EndFrameID, [208](#)
- EventCounter0EndTimestamp, [209](#)
- EventCounter0Start, [209](#)
- EventCounter0StartFrameID, [209](#)
- EventCounter0StartTimestamp, [209](#)
- EventCounter1End, [209](#)
- EventCounter1EndFrameID, [209](#)
- EventCounter1EndTimestamp, [209](#)
- EventCounter1Start, [209](#)
- EventCounter1StartFrameID, [210](#)
- EventCounter1StartTimestamp, [210](#)
- EventEncoder0Restarted, [210](#)
- EventEncoder0RestartedFrameID, [210](#)
- EventEncoder0RestartedTimestamp, [210](#)
- EventEncoder0Stopped, [210](#)
- EventEncoder0StoppedFrameID, [210](#)
- EventEncoder0StoppedTimestamp, [210](#)
- EventEncoder1Restarted, [211](#)
- EventEncoder1RestartedFrameID, [211](#)
- EventEncoder1RestartedTimestamp, [211](#)
- EventEncoder1Stopped, [211](#)
- EventEncoder1StoppedFrameID, [211](#)
- EventEncoder1StoppedTimestamp, [211](#)
- EventError, [211](#)
- EventErrorCode, [211](#)
- EventErrorFrameID, [212](#)
- EventErrorTimestamp, [212](#)
- EventExposureEnd, [212](#)
- EventExposureEndFrameID, [212](#)
- EventExposureEndTimestamp, [212](#)
- EventExposureStart, [212](#)
- EventExposureStartFrameID, [212](#)
- EventExposureStartTimestamp, [212](#)
- EventFrameBurstEnd, [213](#)
- EventFrameBurstEndFrameID, [213](#)
- EventFrameBurstEndTimestamp, [213](#)
- EventFrameBurstStart, [213](#)
- EventFrameBurstStartFrameID, [213](#)
- EventFrameBurstStartTimestamp, [213](#)
- EventFrameEnd, [213](#)
- EventFrameEndFrameID, [213](#)
- EventFrameEndTimestamp, [214](#)
- EventFrameStart, [214](#)
- EventFrameStartFrameID, [214](#)
- EventFrameStartTimestamp, [214](#)
- EventFrameTransferEnd, [214](#)
- EventFrameTransferEndFrameID, [214](#)
- EventFrameTransferEndTimestamp, [214](#)
- EventFrameTransferStart, [214](#)
- EventFrameTransferStartFrameID, [215](#)

- EventFrameTransferStartTimestamp, 215
- EventFrameTrigger, 215
- EventFrameTriggerFrameID, 215
- EventFrameTriggerTimestamp, 215
- EventLine0AnyEdge, 215
- EventLine0AnyEdgeFrameID, 215
- EventLine0AnyEdgeTimestamp, 215
- EventLine0FallingEdge, 216
- EventLine0FallingEdgeFrameID, 216
- EventLine0FallingEdgeTimestamp, 216
- EventLine0RisingEdge, 216
- EventLine0RisingEdgeFrameID, 216
- EventLine0RisingEdgeTimestamp, 216
- EventLine1AnyEdge, 216
- EventLine1AnyEdgeFrameID, 216
- EventLine1AnyEdgeTimestamp, 217
- EventLine1FallingEdge, 217
- EventLine1FallingEdgeFrameID, 217
- EventLine1FallingEdgeTimestamp, 217
- EventLine1RisingEdge, 217
- EventLine1RisingEdgeFrameID, 217
- EventLine1RisingEdgeTimestamp, 217
- EventLinkSpeedChange, 217
- EventLinkSpeedChangeFrameID, 218
- EventLinkSpeedChangeTimestamp, 218
- EventLinkTrigger0, 218
- EventLinkTrigger0FrameID, 218
- EventLinkTrigger0Timestamp, 218
- EventLinkTrigger1, 218
- EventLinkTrigger1FrameID, 218
- EventLinkTrigger1Timestamp, 218
- EventNotification, 219
- EventSelector, 219
- EventSequencerSetChange, 219
- EventSequencerSetChangeFrameID, 219
- EventSequencerSetChangeTimestamp, 219
- EventSerialData, 219
- EventSerialDataLength, 219
- EventSerialPortReceive, 219
- EventSerialPortReceiveTimestamp, 220
- EventSerialReceiveOverflow, 220
- EventStream0TransferBlockEnd, 220
- EventStream0TransferBlockEndFrameID, 220
- EventStream0TransferBlockEndTimestamp, 220
- EventStream0TransferBlockStart, 220
- EventStream0TransferBlockStartFrameID, 220
- EventStream0TransferBlockStartTimestamp, 220
- EventStream0TransferBlockTrigger, 221
- EventStream0TransferBlockTriggerFrameID, 221
- EventStream0TransferBlockTriggerTimestamp, 221
- EventStream0TransferBurstEnd, 221
- EventStream0TransferBurstEndFrameID, 221
- EventStream0TransferBurstEndTimestamp, 221
- EventStream0TransferBurstStart, 221
- EventStream0TransferBurstStartFrameID, 221
- EventStream0TransferBurstStartTimestamp, 222
- EventStream0TransferEnd, 222
- EventStream0TransferEndFrameID, 222
- EventStream0TransferEndTimestamp, 222
- EventStream0TransferOverflow, 222
- EventStream0TransferOverflowFrameID, 222
- EventStream0TransferOverflowTimestamp, 222
- EventStream0TransferPause, 222
- EventStream0TransferPauseFrameID, 223
- EventStream0TransferPauseTimestamp, 223
- EventStream0TransferResume, 223
- EventStream0TransferResumeFrameID, 223
- EventStream0TransferResumeTimestamp, 223
- EventStream0TransferStart, 223
- EventStream0TransferStartFrameID, 223
- EventStream0TransferStartTimestamp, 223
- EventTest, 224
- EventTestTimestamp, 224
- EventTimer0End, 224
- EventTimer0EndFrameID, 224
- EventTimer0EndTimestamp, 224
- EventTimer0Start, 224
- EventTimer0StartFrameID, 224
- EventTimer0StartTimestamp, 224
- EventTimer1End, 225
- EventTimer1EndFrameID, 225
- EventTimer1EndTimestamp, 225
- EventTimer1Start, 225
- EventTimer1StartFrameID, 225
- EventTimer1StartTimestamp, 225
- ExposureActiveMode, 225
- ExposureAuto, 225
- ExposureMode, 226
- ExposureTime, 226
- ExposureTimeMode, 226
- ExposureTimeSelector, 226
- FactoryReset, 226
- FileAccessBuffer, 226
- FileAccessLength, 226
- FileAccessOffset, 226
- FileOpenMode, 227
- FileOperationExecute, 227
- FileOperationResult, 227
- FileOperationSelector, 227
- FileOperationStatus, 227
- FileSelector, 227
- FileSize, 227
- Gain, 227
- GainAuto, 228
- GainAutoBalance, 228
- GainSelector, 228
- Gamma, 228
- GammaEnable, 228
- GevActiveLinkCount, 228
- GevCCP, 228
- GevCurrentDefaultGateway, 228
- GevCurrentIPAddress, 229
- GevCurrentIPConfigurationDHCP, 229
- GevCurrentIPConfigurationLLA, 229
- GevCurrentIPConfigurationPersistentIP, 229
- GevCurrentPhysicalLinkConfiguration, 229

GevCurrentSubnetMask, [229](#)
GevDiscoveryAckDelay, [229](#)
GevFirstURL, [229](#)
GevGVCPExtendedStatusCodes, [230](#)
GevGVCPExtendedStatusCodesSelector, [230](#)
GevGVCPHeartbeatDisable, [230](#)
GevGVCPPendingAck, [230](#)
GevGVCPPendingTimeout, [230](#)
GevGVSPExtendedIDMode, [230](#)
GevHeartbeatTimeout, [230](#)
GevIEEE1588, [230](#)
GevIEEE1588ClockAccuracy, [231](#)
GevIEEE1588Mode, [231](#)
GevIEEE1588Status, [231](#)
GevInterfaceSelector, [231](#)
GevIPConfigurationStatus, [231](#)
GevMACAddress, [231](#)
GevMCDA, [231](#)
GevMCPHostPort, [231](#)
GevMCRC, [232](#)
GevMCSP, [232](#)
GevMCTT, [232](#)
GevNumberOfInterfaces, [232](#)
GevPAUSEFrameReception, [232](#)
GevPAUSEFrameTransmission, [232](#)
GevPersistentDefaultGateway, [232](#)
GevPersistentIPAddress, [232](#)
GevPersistentSubnetMask, [233](#)
GevPhysicalLinkConfiguration, [233](#)
GevPrimaryApplicationIPAddress, [233](#)
GevPrimaryApplicationSocket, [233](#)
GevPrimaryApplicationSwitchoverKey, [233](#)
GevSCCFGAllInTransmission, [233](#)
GevSCCFGExtendedChunkData, [233](#)
GevSCCFGPacketResendDestination, [233](#)
GevSCCFGUnconditionalStreaming, [234](#)
GevSCDA, [234](#)
GevSCPD, [234](#)
GevSCPDirection, [234](#)
GevSCPHostPort, [234](#)
GevSCPInterfaceIndex, [234](#)
GevSCPSBigEndian, [234](#)
GevSCPSDoNotFragment, [234](#)
GevSCPSFireTestPacket, [235](#)
GevSCPSPacketSize, [235](#)
GevSCSP, [235](#)
GevSCZoneConfigurationLock, [235](#)
GevSCZoneCount, [235](#)
GevSCZoneDirectionAll, [235](#)
GevSecondURL, [235](#)
GevStreamChannelSelector, [235](#)
GevSupportedOption, [236](#)
GevSupportedOptionSelector, [236](#)
GevTimestampTickFrequency, [236](#)
GuiXmlManifestAddress, [236](#)
Height, [236](#)
HeightMax, [236](#)
ImageComponentEnable, [236](#)
ImageComponentSelector, [236](#)
ImageCompressionBitrate, [237](#)
ImageCompressionJPEGFormatOption, [237](#)
ImageCompressionMode, [237](#)
ImageCompressionQuality, [237](#)
ImageCompressionRateOption, [237](#)
IspEnable, [237](#)
LineFilterWidth, [237](#)
LineFormat, [237](#)
LineInputFilterSelector, [238](#)
LineInverter, [238](#)
LineMode, [238](#)
LinePitch, [238](#)
LineSelector, [238](#)
LineSource, [238](#)
LineStatus, [238](#)
LineStatusAll, [238](#)
LinkErrorCount, [239](#)
LinkUptime, [239](#)
LogicBlockLUTInputActivation, [239](#)
LogicBlockLUTInputSelector, [239](#)
LogicBlockLUTInputSource, [239](#)
LogicBlockLUTOutputValue, [239](#)
LogicBlockLUTOutputValueAll, [239](#)
LogicBlockLUTRowIndex, [239](#)
LogicBlockLUTSelector, [240](#)
LogicBlockSelector, [240](#)
LUTEnable, [240](#)
LUTIndex, [240](#)
LUTSelector, [240](#)
LUTValue, [240](#)
LUTValueAll, [240](#)
MaxDeviceResetTime, [240](#)
OffsetX, [241](#)
OffsetY, [241](#)
PacketResendRequestCount, [241](#)
PayloadSize, [241](#)
PixelColorFilter, [241](#)
PixelDynamicRangeMax, [241](#)
PixelDynamicRangeMin, [241](#)
PixelFormat, [241](#)
PixelFormatInfoID, [242](#)
PixelFormatInfoSelector, [242](#)
PixelSize, [242](#)
PowerSupplyCurrent, [242](#)
PowerSupplyVoltage, [242](#)
RegionDestination, [242](#)
RegionMode, [242](#)
RegionSelector, [242](#)
ReverseX, [243](#)
ReverseY, [243](#)
RgbTransformLightSource, [243](#)
Saturation, [243](#)
SaturationEnable, [243](#)
Scan3dAxisMax, [243](#)
Scan3dAxisMin, [243](#)
Scan3dCoordinateOffset, [243](#)
Scan3dCoordinateReferenceSelector, [244](#)

Scan3dCoordinateReferenceValue, 244
Scan3dCoordinateScale, 244
Scan3dCoordinateSelector, 244
Scan3dCoordinateSystem, 244
Scan3dCoordinateSystemReference, 244
Scan3dCoordinateTransformSelector, 244
Scan3dDistanceUnit, 244
Scan3dInvalidDataFlag, 245
Scan3dInvalidDataValue, 245
Scan3dOutputMode, 245
Scan3dTransformValue, 245
SensorDescription, 245
SensorDigitizationTaps, 245
SensorHeight, 245
SensorShutterMode, 245
SensorTaps, 246
SensorWidth, 246
SequencerConfigurationMode, 246
SequencerConfigurationValid, 246
SequencerFeatureEnable, 246
SequencerMode, 246
SequencerPathSelector, 246
SequencerSetActive, 246
SequencerSetLoad, 247
SequencerSetNext, 247
SequencerSetSave, 247
SequencerSetSelector, 247
SequencerSetStart, 247
SequencerSetValid, 247
SequencerTriggerActivation, 247
SequencerTriggerSource, 247
SerialPortBaudRate, 248
SerialPortDataBits, 248
SerialPortParity, 248
SerialPortSelector, 248
SerialPortSource, 248
SerialPortStopBits, 248
SerialReceiveFramingErrorCount, 248
SerialReceiveParityErrorCount, 248
SerialReceiveQueueClear, 249
SerialReceiveQueueCurrentCharacterCount, 249
SerialReceiveQueueMaxCharacterCount, 249
SerialTransmitQueueCurrentCharacterCount, 249
SerialTransmitQueueMaxCharacterCount, 249
Sharpening, 249
SharpeningAuto, 249
SharpeningEnable, 249
SharpeningThreshold, 250
SoftwareSignalPulse, 250
SoftwareSignalSelector, 250
SourceCount, 250
SourceSelector, 250
Test0001, 250
TestEventGenerate, 250
TestPattern, 250
TestPatternGeneratorSelector, 251
TestPendingAck, 251
TimerDelay, 251
TimerDuration, 251
TimerReset, 251
TimerSelector, 251
TimerStatus, 251
TimerTriggerActivation, 251
TimerTriggerSource, 252
TimerValue, 252
Timestamp, 252
TimestampLatch, 252
TimestampLatchValue, 252
TimestampReset, 252
TLParamsLocked, 252
TransferAbort, 252
TransferBlockCount, 253
TransferBurstCount, 253
TransferComponentSelector, 253
TransferControlMode, 253
TransferOperationMode, 253
TransferPause, 253
TransferQueueCurrentBlockCount, 253
TransferQueueMaxBlockCount, 253
TransferQueueMode, 254
TransferQueueOverflowCount, 254
TransferResume, 254
TransferSelector, 254
TransferStart, 254
TransferStatus, 254
TransferStatusSelector, 254
TransferStop, 254
TransferStreamChannel, 255
TransferTriggerActivation, 255
TransferTriggerMode, 255
TransferTriggerSelector, 255
TransferTriggerSource, 255
TriggerActivation, 255
TriggerDelay, 255
TriggerDivider, 255
TriggerEventTest, 256
TriggerMode, 256
TriggerMultiplier, 256
TriggerOverlap, 256
TriggerSelector, 256
TriggerSoftware, 256
TriggerSource, 256
UserOutputSelector, 256
UserOutputValue, 257
UserOutputValueAll, 257
UserOutputValueAllMask, 257
UserSetDefault, 257
UserSetFeatureEnable, 257
UserSetLoad, 257
UserSetSave, 257
UserSetSelector, 257
V3_3Enable, 258
WhiteClip, 258
WhiteClipSelector, 258
Width, 258
WidthMax, 258

- QuickSpin Access, [142](#)
- quickSpinBooleanNode
 - QuickSpinDefsC.h, [339](#)
- QuickSpinC.h
 - quickSpinInit, [337](#)
 - quickSpinInitEx, [337](#)
 - quickSpinTLDeviceInit, [338](#)
 - quickSpinTLInterfaceInit, [338](#)
 - quickSpinTLStreamInit, [338](#)
 - quickSpinTLSystemInit, [338](#)
- quickSpinCommandNode
 - QuickSpinDefsC.h, [340](#)
- QuickSpinDefsC.h
 - quickSpinBooleanNode, [339](#)
 - quickSpinCommandNode, [340](#)
 - quickSpinEnumerationNode, [340](#)
 - quickSpinFloatNode, [340](#)
 - quickSpinIntegerNode, [340](#)
 - quickSpinRegisterNode, [340](#)
 - quickSpinStringNode, [340](#)
- quickSpinEnumerationNode
 - QuickSpinDefsC.h, [340](#)
- quickSpinFloatNode
 - QuickSpinDefsC.h, [340](#)
- quickSpinInit
 - QuickSpinC.h, [337](#)
- quickSpinInitEx
 - QuickSpinC.h, [337](#)
- quickSpinIntegerNode
 - QuickSpinDefsC.h, [340](#)
- quickSpinRegisterNode
 - QuickSpinDefsC.h, [340](#)
- quickSpinStringNode
 - QuickSpinDefsC.h, [340](#)
- quickSpinTLDevice, [259](#)
 - DeviceAccessStatus, [259](#)
 - DeviceBootloaderVersion, [260](#)
 - DeviceCurrentSpeed, [260](#)
 - DeviceDisplayName, [260](#)
 - DeviceDriverVersion, [260](#)
 - DeviceEndianessMechanism, [260](#)
 - DeviceID, [260](#)
 - DeviceInstanceld, [260](#)
 - DevicesUpdater, [260](#)
 - DeviceLinkSpeed, [261](#)
 - DeviceLocation, [261](#)
 - DeviceModelName, [261](#)
 - DeviceMulticastMonitorMode, [261](#)
 - DevicePortId, [261](#)
 - DeviceReset, [261](#)
 - DeviceSerialNumber, [261](#)
 - DeviceType, [261](#)
 - DeviceU3VProtocol, [262](#)
 - DeviceUserID, [262](#)
 - DeviceVendorName, [262](#)
 - DeviceVersion, [262](#)
 - GenICamXMLLocation, [262](#)
 - GenICamXMLPath, [262](#)
- GevCCP, [262](#)
- GevDeviceAutoForceIP, [262](#)
- GevDeviceDiscoverMaximumPacketSize, [263](#)
- GevDeviceForceGateway, [263](#)
- GevDeviceForceIP, [263](#)
- GevDeviceForceIPAddress, [263](#)
- GevDeviceForceSubnetMask, [263](#)
- GevDeviceGateway, [263](#)
- GevDeviceIPAddress, [263](#)
- GevDevicesWrongSubnet, [263](#)
- GevDeviceMACAddress, [264](#)
- GevDeviceMaximumPacketSize, [264](#)
- GevDeviceMaximumRetryCount, [264](#)
- GevDeviceModelsBigEndian, [264](#)
- GevDevicePort, [264](#)
- GevDeviceReadAndWriteTimeout, [264](#)
- GevDeviceSubnetMask, [264](#)
- GevVersionMajor, [264](#)
- GevVersionMinor, [265](#)
- GUIXMLLocation, [265](#)
- GUIXMLPath, [265](#)
- quickSpinTLDeviceInit
 - QuickSpinC.h, [338](#)
- quickSpinTLInterface, [265](#)
 - ActionCommand, [266](#)
 - DeviceAccessStatus, [266](#)
 - DeviceCount, [266](#)
 - DeviceID, [267](#)
 - DeviceModelName, [267](#)
 - DeviceSelector, [267](#)
 - DeviceSerialNumber, [267](#)
 - DeviceUnlock, [267](#)
 - DeviceUpdateList, [267](#)
 - DeviceVendorName, [267](#)
 - FLIRFilterDriverStatus, [267](#)
 - GevActionAckRequired, [268](#)
 - GevActionDeviceKey, [268](#)
 - GevActionGroupKey, [268](#)
 - GevActionGroupMask, [268](#)
 - GevActionTime, [268](#)
 - GevDeviceAutoForceIP, [268](#)
 - GevDeviceDisableDiscovery, [268](#)
 - GevDeviceDiscoveryEnabled, [268](#)
 - GevDeviceEnableDiscovery, [269](#)
 - GevDeviceForceGateway, [269](#)
 - GevDeviceForceIP, [269](#)
 - GevDeviceForceIPAddress, [269](#)
 - GevDeviceForceSubnetMask, [269](#)
 - GevDeviceGateway, [269](#)
 - GevDeviceIPAddress, [269](#)
 - GevDeviceMACAddress, [269](#)
 - GevDeviceSubnetMask, [270](#)
 - GevInterfaceGateway, [270](#)
 - GevInterfaceGatewaySelector, [270](#)
 - GevInterfaceIPConflict, [270](#)
 - GevInterfaceMACAddress, [270](#)
 - GevInterfaceMTU, [270](#)
 - GevInterfaceReceiveLinkSpeed, [270](#)

- GevInterfaceSubnetIPAddress, [270](#)
- GevInterfaceSubnetMask, [271](#)
- GevInterfaceSubnetSelector, [271](#)
- GevInterfaceTransmitLinkSpeed, [271](#)
- HostAdapterDriverVersion, [271](#)
- HostAdapterName, [271](#)
- HostAdapterVendor, [271](#)
- IncompatibleDeviceCount, [271](#)
- IncompatibleDeviceID, [271](#)
- IncompatibleDeviceModelName, [272](#)
- IncompatibleDeviceSelector, [272](#)
- IncompatibleDeviceVendorName, [272](#)
- IncompatibleGevDeviceIPAddress, [272](#)
- IncompatibleGevDeviceMACAddress, [272](#)
- IncompatibleGevDeviceSubnetMask, [272](#)
- InterfaceDisplayName, [272](#)
- InterfaceID, [272](#)
- InterfaceType, [273](#)
- POEStatus, [273](#)
- TeledyneGigeVisionFilterDriverStatus, [273](#)
- quickSpinTLInterfaceInit
 - QuickSpinC.h, [338](#)
- quickSpinTLStream, [273](#)
 - StreamAnnounceBufferMinimum, [274](#)
 - StreamAnnouncedBufferCount, [274](#)
 - StreamBlocksProcessingTimeLast, [274](#)
 - StreamBlocksProcessingTimeMax, [274](#)
 - StreamBlocksProcessingTimeMin, [275](#)
 - StreamBlocksReceptionTimeLast, [275](#)
 - StreamBlocksReceptionTimeMax, [275](#)
 - StreamBlocksReceptionTimeMin, [275](#)
 - StreamBlockTransferSize, [275](#)
 - StreamBufferAlignment, [275](#)
 - StreamBufferCountManual, [275](#)
 - StreamBufferCountMax, [275](#)
 - StreamBufferCountMode, [276](#)
 - StreamBufferCountResult, [276](#)
 - StreamBufferHandlingMode, [276](#)
 - StreamChunkCountMaximum, [276](#)
 - StreamCRCCheckEnable, [276](#)
 - StreamDeliveredFrameCount, [276](#)
 - StreamDroppedFrameCount, [276](#)
 - StreamID, [276](#)
 - StreamIncompleteFrameCount, [277](#)
 - StreamInputBufferCount, [277](#)
 - StreamIsGrabbing, [277](#)
 - StreamLostFrameCount, [277](#)
 - StreamMissedPacketCount, [277](#)
 - StreamMode, [277](#)
 - StreamOutputBufferCount, [277](#)
 - StreamPacketResendEnable, [277](#)
 - StreamPacketResendMaxRequests, [278](#)
 - StreamPacketResendReceivedPacketCount, [278](#)
 - StreamPacketResendRequestCount, [278](#)
 - StreamPacketResendRequestedPacketCount, [278](#)
 - StreamPacketResendRequestTimeoutCount, [278](#)
 - StreamPacketResendTimeout, [278](#)
 - StreamPacketsDuplicatedCount, [278](#)
 - StreamPacketsNotYetAvailableCount, [278](#)
 - StreamPacketsPerFrameCount, [279](#)
 - StreamPacketsTemporarilyUnavailableCount, [279](#)
 - StreamPacketsTimeoutCount, [279](#)
 - StreamPacketsUnavailableCount, [279](#)
 - StreamReceivedFrameCount, [279](#)
 - StreamReceivedPacketCount, [279](#)
 - StreamStartedFrameCount, [279](#)
 - StreamType, [279](#)
- quickSpinTLStreamInit
 - QuickSpinC.h, [338](#)
- quickSpinTLSystem, [280](#)
 - EnumerateGen2Cameras, [280](#)
 - EnumerateGEVInterfaces, [280](#)
 - EnumerateUSBInterfaces, [281](#)
 - GenTLSENCVersionMajor, [281](#)
 - GenTLSENCVersionMinor, [281](#)
 - GenTLSENCVersionSubMinor, [281](#)
 - GenTLVersionMajor, [281](#)
 - GenTLVersionMinor, [281](#)
 - GevAutoAssignIPEnable, [281](#)
 - GevInterfaceDefaultGateway, [281](#)
 - GevInterfaceDefaultIPAddress, [282](#)
 - GevInterfaceDefaultSubnetMask, [282](#)
 - GevInterfaceMACAddress, [282](#)
 - GevVersionMajor, [282](#)
 - GevVersionMinor, [282](#)
 - InterfaceDisplayName, [282](#)
 - InterfaceID, [282](#)
 - InterfaceSelector, [282](#)
 - InterfaceUpdateList, [283](#)
 - TLDisplayName, [283](#)
 - TLFileName, [283](#)
 - TLID, [283](#)
 - TLModelName, [283](#)
 - TLPath, [283](#)
 - TLType, [283](#)
 - TLVendorName, [283](#)
 - TLVersion, [284](#)
- quickSpinTLSystemInit
 - QuickSpinC.h, [338](#)
- RegionDestination
 - quickSpin, [242](#)
- RegionDestination_Stream0
 - Camera Enumerations, [119](#)
- RegionDestination_Stream1
 - Camera Enumerations, [119](#)
- RegionDestination_Stream2
 - Camera Enumerations, [119](#)
- RegionMode
 - quickSpin, [242](#)
- RegionMode_Off
 - Camera Enumerations, [119](#)
- RegionMode_On
 - Camera Enumerations, [119](#)
- RegionSelector
 - quickSpin, [242](#)
- RegionSelector_All

- Camera Enumerations, [120](#)
- RegionSelector_Region0
 - Camera Enumerations, [120](#)
- RegionSelector_Region1
 - Camera Enumerations, [120](#)
- RegionSelector_Region2
 - Camera Enumerations, [120](#)
- RegisterNode
 - SpinnakerGenApiDefsC.h, [511](#)
- reserved
 - spinAVIOption, [285](#)
 - spinBMPOption, [286](#)
 - spinH264Option, [293](#)
 - spinJPEGOption, [294](#)
 - spinJPG2Option, [295](#)
 - spinMJPEGOption, [298](#)
 - spinPGMOption, [299](#)
 - spinPNGOption, [300](#)
 - spinPPMOption, [301](#)
 - spinTIFFOption, [301](#)
- ReverseX
 - quickSpin, [243](#)
- ReverseY
 - quickSpin, [243](#)
- RgbTransformLightSource
 - quickSpin, [243](#)
- RgbTransformLightSource_Cloudy6500K
 - Camera Enumerations, [120](#)
- RgbTransformLightSource_CoolFluorescent4000K
 - Camera Enumerations, [120](#)
- RgbTransformLightSource_Custom
 - Camera Enumerations, [120](#)
- RgbTransformLightSource_Daylight5000K
 - Camera Enumerations, [120](#)
- RgbTransformLightSource_General
 - Camera Enumerations, [120](#)
- RgbTransformLightSource_Shade8000K
 - Camera Enumerations, [120](#)
- RgbTransformLightSource_Tungsten2800K
 - Camera Enumerations, [120](#)
- RgbTransformLightSource_WarmFluorescent3000K
 - Camera Enumerations, [120](#)
- RO
 - SpinnakerGenApiDefsC.h, [507](#)
- RW
 - SpinnakerGenApiDefsC.h, [507](#)
- Saturation
 - quickSpin, [243](#)
- SaturationEnable
 - quickSpin, [243](#)
- Scan3dAxisMax
 - quickSpin, [243](#)
- Scan3dAxisMin
 - quickSpin, [243](#)
- Scan3dCoordinateOffset
 - quickSpin, [243](#)
- Scan3dCoordinateReferenceSelector
 - quickSpin, [244](#)
- Scan3dCoordinateReferenceSelector_RotationX
 - Camera Enumerations, [121](#)
- Scan3dCoordinateReferenceSelector_RotationY
 - Camera Enumerations, [121](#)
- Scan3dCoordinateReferenceSelector_RotationZ
 - Camera Enumerations, [121](#)
- Scan3dCoordinateReferenceSelector_TranslationX
 - Camera Enumerations, [121](#)
- Scan3dCoordinateReferenceSelector_TranslationY
 - Camera Enumerations, [121](#)
- Scan3dCoordinateReferenceSelector_TranslationZ
 - Camera Enumerations, [121](#)
- Scan3dCoordinateReferenceValue
 - quickSpin, [244](#)
- Scan3dCoordinateScale
 - quickSpin, [244](#)
- Scan3dCoordinateSelector
 - quickSpin, [244](#)
- Scan3dCoordinateSelector_CoordinateA
 - Camera Enumerations, [121](#)
- Scan3dCoordinateSelector_CoordinateB
 - Camera Enumerations, [121](#)
- Scan3dCoordinateSelector_CoordinateC
 - Camera Enumerations, [121](#)
- Scan3dCoordinateSystem
 - quickSpin, [244](#)
- Scan3dCoordinateSystem_Cartesian
 - Camera Enumerations, [121](#)
- Scan3dCoordinateSystem_Cylindrical
 - Camera Enumerations, [121](#)
- Scan3dCoordinateSystem_Spherical
 - Camera Enumerations, [121](#)
- Scan3dCoordinateSystemReference
 - quickSpin, [244](#)
- Scan3dCoordinateSystemReference_Anchor
 - Camera Enumerations, [122](#)
- Scan3dCoordinateSystemReference_Transformed
 - Camera Enumerations, [122](#)
- Scan3dCoordinateTransformSelector
 - quickSpin, [244](#)
- Scan3dCoordinateTransformSelector_RotationX
 - Camera Enumerations, [122](#)
- Scan3dCoordinateTransformSelector_RotationY
 - Camera Enumerations, [122](#)
- Scan3dCoordinateTransformSelector_RotationZ
 - Camera Enumerations, [122](#)
- Scan3dCoordinateTransformSelector_TranslationX
 - Camera Enumerations, [122](#)
- Scan3dCoordinateTransformSelector_TranslationY
 - Camera Enumerations, [122](#)
- Scan3dCoordinateTransformSelector_TranslationZ
 - Camera Enumerations, [122](#)
- Scan3dDistanceUnit
 - quickSpin, [244](#)
- Scan3dDistanceUnit_Inch
 - Camera Enumerations, [122](#)
- Scan3dDistanceUnit_Millimeter
 - Camera Enumerations, [122](#)

- Scan3dInvalidDataFlag
 - quickSpin, [245](#)
- Scan3dInvalidDataValue
 - quickSpin, [245](#)
- Scan3dOutputMode
 - quickSpin, [245](#)
- Scan3dOutputMode_CalibratedABC_Grid
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_CalibratedABC_PointCloud
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_CalibratedAC
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_CalibratedAC_Linescan
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_CalibratedC
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_CalibratedC_Linescan
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_DisparityC
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_DisparityC_Linescan
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_RectifiedC
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_RectifiedC_Linescan
 - Camera Enumerations, [124](#)
- Scan3dOutputMode_UncalibratedC
 - Camera Enumerations, [124](#)
- Scan3dTransformValue
 - quickSpin, [245](#)
- SensorDescription
 - quickSpin, [245](#)
- SensorDigitizationTaps
 - quickSpin, [245](#)
- SensorDigitizationTaps_Eight
 - Camera Enumerations, [125](#)
- SensorDigitizationTaps_Four
 - Camera Enumerations, [125](#)
- SensorDigitizationTaps_One
 - Camera Enumerations, [124](#)
- SensorDigitizationTaps_Ten
 - Camera Enumerations, [125](#)
- SensorDigitizationTaps_Three
 - Camera Enumerations, [124](#)
- SensorDigitizationTaps_Two
 - Camera Enumerations, [124](#)
- SensorHeight
 - quickSpin, [245](#)
- SensorShutterMode
 - quickSpin, [245](#)
- SensorShutterMode_Global
 - Camera Enumerations, [125](#)
- SensorShutterMode_GlobalReset
 - Camera Enumerations, [125](#)
- SensorShutterMode_Rolling
 - Camera Enumerations, [125](#)
- SensorTaps
 - quickSpin, [246](#)
- SensorTaps_Eight
 - Camera Enumerations, [125](#)
- SensorTaps_Four
 - Camera Enumerations, [125](#)
- SensorTaps_One
 - Camera Enumerations, [125](#)
- SensorTaps_Ten
 - Camera Enumerations, [125](#)
- SensorTaps_Three
 - Camera Enumerations, [125](#)
- SensorTaps_Two
 - Camera Enumerations, [125](#)
- SensorWidth
 - quickSpin, [246](#)
- SequencerConfigurationMode
 - quickSpin, [246](#)
- SequencerConfigurationMode_Off
 - Camera Enumerations, [126](#)
- SequencerConfigurationMode_On
 - Camera Enumerations, [126](#)
- SequencerConfigurationValid
 - quickSpin, [246](#)
- SequencerConfigurationValid_No
 - Camera Enumerations, [126](#)
- SequencerConfigurationValid_Yes
 - Camera Enumerations, [126](#)
- SequencerFeatureEnable
 - quickSpin, [246](#)
- SequencerMode
 - quickSpin, [246](#)
- SequencerMode_Off
 - Camera Enumerations, [126](#)
- SequencerMode_On
 - Camera Enumerations, [126](#)
- SequencerPathSelector
 - quickSpin, [246](#)
- SequencerSetActive
 - quickSpin, [246](#)
- SequencerSetLoad
 - quickSpin, [247](#)
- SequencerSetNext
 - quickSpin, [247](#)
- SequencerSetSave
 - quickSpin, [247](#)
- SequencerSetSelector
 - quickSpin, [247](#)
- SequencerSetStart
 - quickSpin, [247](#)
- SequencerSetValid
 - quickSpin, [247](#)
- SequencerSetValid_No
 - Camera Enumerations, [127](#)
- SequencerSetValid_Yes
 - Camera Enumerations, [127](#)
- SequencerTriggerActivation
 - quickSpin, [247](#)
- SequencerTriggerActivation_AnyEdge
 - Camera Enumerations, [127](#)

- SequencerTriggerActivation_FallingEdge
 - Camera Enumerations, [127](#)
- SequencerTriggerActivation_LevelHigh
 - Camera Enumerations, [127](#)
- SequencerTriggerActivation_LevelLow
 - Camera Enumerations, [127](#)
- SequencerTriggerActivation_RisingEdge
 - Camera Enumerations, [127](#)
- SequencerTriggerSource
 - quickSpin, [247](#)
- SequencerTriggerSource_FrameStart
 - Camera Enumerations, [127](#)
- SequencerTriggerSource_Off
 - Camera Enumerations, [127](#)
- SerialPortBaudRate
 - quickSpin, [248](#)
- SerialPortBaudRate_Baud115200
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud1200
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud14400
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud19200
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud230400
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud2400
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud300
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud38400
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud460800
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud4800
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud57600
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud600
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud921600
 - Camera Enumerations, [128](#)
- SerialPortBaudRate_Baud9600
 - Camera Enumerations, [128](#)
- SerialPortDataBits
 - quickSpin, [248](#)
- SerialPortParity
 - quickSpin, [248](#)
- SerialPortParity_Even
 - Camera Enumerations, [128](#)
- SerialPortParity_Mark
 - Camera Enumerations, [128](#)
- SerialPortParity_None
 - Camera Enumerations, [128](#)
- SerialPortParity_Odd
 - Camera Enumerations, [128](#)
- SerialPortParity_Space
 - Camera Enumerations, [128](#)
- SerialPortSelector
 - quickSpin, [248](#)
- SerialPortSelector_SerialPort0
 - Camera Enumerations, [128](#)
- SerialPortSource
 - quickSpin, [248](#)
- SerialPortSource_Line0
 - Camera Enumerations, [129](#)
- SerialPortSource_Line1
 - Camera Enumerations, [129](#)
- SerialPortSource_Line2
 - Camera Enumerations, [129](#)
- SerialPortSource_Line3
 - Camera Enumerations, [129](#)
- SerialPortSource_Off
 - Camera Enumerations, [129](#)
- SerialPortStopBits
 - quickSpin, [248](#)
- SerialPortStopBits_Bits1
 - Camera Enumerations, [129](#)
- SerialPortStopBits_Bits1AndAHalf
 - Camera Enumerations, [129](#)
- SerialPortStopBits_Bits2
 - Camera Enumerations, [129](#)
- SerialReceiveFramingErrorCount
 - quickSpin, [248](#)
- SerialReceiveParityErrorCount
 - quickSpin, [248](#)
- SerialReceiveQueueClear
 - quickSpin, [249](#)
- SerialReceiveQueueCurrentCharacterCount
 - quickSpin, [249](#)
- SerialReceiveQueueMaxCharacterCount
 - quickSpin, [249](#)
- SerialTransmitQueueCurrentCharacterCount
 - quickSpin, [249](#)
- SerialTransmitQueueMaxCharacterCount
 - quickSpin, [249](#)
- Sharpening
 - quickSpin, [249](#)
- SharpeningAuto
 - quickSpin, [249](#)
- SharpeningEnable
 - quickSpin, [249](#)
- SharpeningThreshold
 - quickSpin, [250](#)
- Signed
 - SpinnakerGenApiDefsC.h, [511](#)
- SoftwareSignalPulse
 - quickSpin, [250](#)
- SoftwareSignalSelector
 - quickSpin, [250](#)
- SoftwareSignalSelector_SoftwareSignal0
 - Camera Enumerations, [129](#)
- SoftwareSignalSelector_SoftwareSignal1
 - Camera Enumerations, [129](#)
- SoftwareSignalSelector_SoftwareSignal2
 - Camera Enumerations, [129](#)

- SourceCount
 - quickSpin, [250](#)
- SourceSelector
 - quickSpin, [250](#)
- SourceSelector_All
 - Camera Enumerations, [130](#)
- SourceSelector_Source0
 - Camera Enumerations, [130](#)
- SourceSelector_Source1
 - Camera Enumerations, [130](#)
- SourceSelector_Source2
 - Camera Enumerations, [130](#)
- spinAccessMode
 - SpinnakerGenApiDefsC.h, [506](#)
- spinAcquisitionModeEnums
 - Camera Enumerations, [56](#)
- spinAcquisitionStatusSelectorEnums
 - Camera Enumerations, [56](#)
- spinActionCommandStatus
 - SpinnakerDefsC.h, [456](#)
- spinActionUnconditionalModeEnums
 - Camera Enumerations, [57](#)
- spinAdcBitDepthEnums
 - Camera Enumerations, [57](#)
- spinArrivalEventFunction
 - SpinnakerDefsC.h, [452](#)
- spinAutoAlgorithmSelectorEnums
 - Camera Enumerations, [57](#)
- spinAutoExposureControlPriorityEnums
 - Camera Enumerations, [58](#)
- spinAutoExposureLightingModeEnums
 - Camera Enumerations, [58](#)
- spinAutoExposureMeteringModeEnums
 - Camera Enumerations, [59](#)
- spinAutoExposureTargetGreyValueAutoEnums
 - Camera Enumerations, [59](#)
- spinAVIOption, [284](#)
 - frameRate, [284](#)
 - height, [284](#)
 - reserved, [285](#)
 - width, [285](#)
- spinBalanceRatioSelectorEnums
 - Camera Enumerations, [59](#)
- spinBalanceWhiteAutoEnums
 - Camera Enumerations, [60](#)
- spinBalanceWhiteAutoProfileEnums
 - Camera Enumerations, [60](#)
- spinBinningHorizontalModeEnums
 - Camera Enumerations, [60](#)
- spinBinningSelectorEnums
 - Camera Enumerations, [61](#)
- spinBinningVerticalModeEnums
 - Camera Enumerations, [61](#)
- spinBlackLevelAutoBalanceEnums
 - Camera Enumerations, [61](#)
- spinBlackLevelAutoEnums
 - Camera Enumerations, [62](#)
- spinBlackLevelSelectorEnums
 - Camera Enumerations, [62](#)
- spinBMPOption, [285](#)
 - indexedColor_8bit, [285](#)
 - reserved, [286](#)
- spinBooleanGetValue
 - SpinnakerGenApiC.h, [466](#)
- spinBooleanSetValue
 - SpinnakerGenApiC.h, [467](#)
- spinCachingMode
 - SpinnakerGenApiDefsC.h, [507](#)
- spinCamera
 - SpinnakerDefsC.h, [452](#)
- spinCameraBeginAcquisition
 - SpinnakerC.h, [351](#)
- spinCameraDelInit
 - SpinnakerC.h, [351](#)
- spinCameraDiscoverMaxPacketSize
 - SpinnakerC.h, [351](#)
- spinCameraEndAcquisition
 - SpinnakerC.h, [352](#)
- spinCameraForceIP
 - SpinnakerC.h, [352](#)
- spinCameraGetAccessMode
 - SpinnakerC.h, [353](#)
- spinCameraGetDeviceID
 - SpinnakerC.h, [353](#)
- spinCameraGetGuiXml
 - SpinnakerC.h, [354](#)
- spinCameraGetNextImage
 - SpinnakerC.h, [354](#)
- spinCameraGetNextImageEx
 - SpinnakerC.h, [355](#)
- spinCameraGetNextImageSync
 - SpinnakerC.h, [355](#)
- spinCameraGetNodeMap
 - SpinnakerC.h, [356](#)
- spinCameraGetTLDeviceNodeMap
 - SpinnakerC.h, [356](#)
- spinCameraGetTLStreamNodeMap
 - SpinnakerC.h, [357](#)
- spinCameraInit
 - SpinnakerC.h, [357](#)
- spinCamerasInitialized
 - SpinnakerC.h, [358](#)
- spinCamerasValid
 - SpinnakerC.h, [358](#)
- spinCameraList
 - SpinnakerDefsC.h, [452](#)
- spinCameraListAppend
 - SpinnakerC.h, [359](#)
- spinCameraListClear
 - SpinnakerC.h, [359](#)
- spinCameraListCreateEmpty
 - SpinnakerC.h, [359](#)
- spinCameraListDestroy
 - SpinnakerC.h, [360](#)
- spinCameraListGet
 - SpinnakerC.h, [360](#)

- spinCameraListGetBySerial
 - SpinnakerC.h, [361](#)
- spinCameraListGetSize
 - SpinnakerC.h, [361](#)
- spinCameraListRemove
 - SpinnakerC.h, [362](#)
- spinCameraListRemoveBySerial
 - SpinnakerC.h, [362](#)
- spinCameraReadPort
 - SpinnakerC.h, [363](#)
- spinCameraRegisterDeviceEventHandler
 - SpinnakerC.h, [363](#)
- spinCameraRegisterDeviceEventHandlerEx
 - SpinnakerC.h, [363](#)
- spinCameraRegisterImageEventHandler
 - SpinnakerC.h, [364](#)
- spinCameraRegisterImageEventHandlerEx
 - SpinnakerC.h, [364](#)
- spinCameraRegisterImageListEventHandler
 - SpinnakerC.h, [365](#)
- spinCameraRelease
 - SpinnakerC.h, [365](#)
- spinCameraUnregisterDeviceEventHandler
 - SpinnakerC.h, [366](#)
- spinCameraUnregisterImageEventHandler
 - SpinnakerC.h, [366](#)
- spinCameraUnregisterImageListEventHandler
 - SpinnakerC.h, [367](#)
- spinCameraWritePort
 - SpinnakerC.h, [367](#)
- spinCategoryGetFeatureByIndex
 - SpinnakerGenApiC.h, [467](#)
- spinCategoryGetNumFeatures
 - SpinnakerGenApiC.h, [468](#)
- spinCategoryReleaseNode
 - SpinnakerGenApiC.h, [468](#)
- spinChunkBlackLevelSelectorEnums
 - Camera Enumerations, [62](#)
- spinChunkCounterSelectorEnums
 - Camera Enumerations, [63](#)
- spinChunkData, [286](#)
 - m_blackLevel, [287](#)
 - m_compressionMode, [287](#)
 - m_compressionRatio, [287](#)
 - m_counterValue, [287](#)
 - m_cRC, [287](#)
 - m_encoderValue, [288](#)
 - m_exposureEndLineStatusAll, [288](#)
 - m_exposureTime, [288](#)
 - m_frameID, [288](#)
 - m_gain, [288](#)
 - m_height, [288](#)
 - m_image, [288](#)
 - m_inferenceConfidence, [288](#)
 - m_inferenceFrameID, [289](#)
 - m_inferenceResult, [289](#)
 - m_linePitch, [289](#)
 - m_lineStatusAll, [289](#)
 - m_offsetX, [289](#)
 - m_offsetY, [289](#)
 - m_partSelector, [289](#)
 - m_pixelDynamicRangeMax, [289](#)
 - m_pixelDynamicRangeMin, [290](#)
 - m_scan3dAxisMax, [290](#)
 - m_scan3dAxisMin, [290](#)
 - m_scan3dCoordinateOffset, [290](#)
 - m_scan3dCoordinateReferenceValue, [290](#)
 - m_scan3dCoordinateScale, [290](#)
 - m_scan3dInvalidDataValue, [290](#)
 - m_scan3dTransformValue, [290](#)
 - m_scanLineSelector, [291](#)
 - m_sequencerSetActive, [291](#)
 - m_serialDataLength, [291](#)
 - m_streamChannelID, [291](#)
 - m_timerValue, [291](#)
 - m_timestamp, [291](#)
 - m_timestampLatchValue, [291](#)
 - m_transferBlockID, [291](#)
 - m_transferQueueCurrentBlockCount, [292](#)
 - m_width, [292](#)
- spinChunkEncoderSelectorEnums
 - Camera Enumerations, [63](#)
- spinChunkEncoderStatusEnums
 - Camera Enumerations, [63](#)
- spinChunkExposureTimeSelectorEnums
 - Camera Enumerations, [64](#)
- spinChunkGainSelectorEnums
 - Camera Enumerations, [64](#)
- spinChunkImageComponentEnums
 - Camera Enumerations, [64](#)
- spinChunkPixelFormatEnums
 - Camera Enumerations, [65](#)
- spinChunkRegionIDEnums
 - Camera Enumerations, [65](#)
- spinChunkScan3dCoordinateReferenceSelectorEnums
 - Camera Enumerations, [66](#)
- spinChunkScan3dCoordinateSelectorEnums
 - Camera Enumerations, [66](#)
- spinChunkScan3dCoordinateSystemEnums
 - Camera Enumerations, [66](#)
- spinChunkScan3dCoordinateSystemReferenceEnums
 - Camera Enumerations, [67](#)
- spinChunkScan3dCoordinateTransformSelectorEnums
 - Camera Enumerations, [67](#)
- spinChunkScan3dDistanceUnitEnums
 - Camera Enumerations, [67](#)
- spinChunkScan3dOutputModeEnums
 - Camera Enumerations, [68](#)
- spinChunkSelectorEnums
 - Camera Enumerations, [68](#)
- spinChunkSourceIDEnums
 - Camera Enumerations, [69](#)
- spinChunkTimerSelectorEnums
 - Camera Enumerations, [69](#)
- spinChunkTransferStreamIDEnums
 - Camera Enumerations, [70](#)

- spinCIConfigurationEnums
 - Camera Enumerations, [70](#)
- spinCITimeSlotsCountEnums
 - Camera Enumerations, [70](#)
- spinColorProcessingAlgorithm
 - SpinnakerDefsC.h, [456](#)
- spinColorTransformationSelectorEnums
 - Camera Enumerations, [71](#)
- spinColorTransformationValueSelectorEnums
 - Camera Enumerations, [71](#)
- spinCommandExecute
 - SpinnakerGenApiC.h, [469](#)
- spinCommandIsDone
 - SpinnakerGenApiC.h, [469](#)
- spinCompressionSaturationPriorityEnums
 - Camera Enumerations, [72](#)
- spinCounterEventActivationEnums
 - Camera Enumerations, [72](#)
- spinCounterEventSourceEnums
 - Camera Enumerations, [72](#)
- spinCounterResetActivationEnums
 - Camera Enumerations, [73](#)
- spinCounterResetSourceEnums
 - Camera Enumerations, [73](#)
- spinCounterSelectorEnums
 - Camera Enumerations, [74](#)
- spinCounterStatusEnums
 - Camera Enumerations, [74](#)
- spinCounterTriggerActivationEnums
 - Camera Enumerations, [75](#)
- spinCounterTriggerSourceEnums
 - Camera Enumerations, [75](#)
- spinCxpConnectionTestModeEnums
 - Camera Enumerations, [76](#)
- spinCxpLinkConfigurationEnums
 - Camera Enumerations, [76](#)
- spinCxpLinkConfigurationPreferredEnums
 - Camera Enumerations, [77](#)
- spinCxpLinkConfigurationStatusEnums
 - Camera Enumerations, [78](#)
- spinCxpPoCxpStatusEnums
 - Camera Enumerations, [79](#)
- spinDecimationHorizontalModeEnums
 - Camera Enumerations, [79](#)
- spinDecimationSelectorEnums
 - Camera Enumerations, [79](#)
- spinDecimationVerticalModeEnums
 - Camera Enumerations, [80](#)
- spinDefectCorrectionModeEnums
 - Camera Enumerations, [80](#)
- spinDeinterlacingEnums
 - Camera Enumerations, [80](#)
- spinDeviceArrivalEventHandler
 - SpinnakerDefsC.h, [452](#)
- spinDeviceArrivalEventHandlerCreate
 - SpinnakerC.h, [367](#)
- spinDeviceArrivalEventHandlerDestroy
 - SpinnakerC.h, [368](#)
- spinDeviceCharacterSetEnums
 - Camera Enumerations, [81](#)
- spinDeviceClockSelectorEnums
 - Camera Enumerations, [81](#)
- spinDeviceConnectionStatusEnums
 - Camera Enumerations, [81](#)
- spinDeviceEventData
 - SpinnakerDefsC.h, [452](#)
- spinDeviceEventFunction
 - SpinnakerDefsC.h, [452](#)
- spinDeviceEventGetId
 - SpinnakerC.h, [368](#)
- spinDeviceEventGetName
 - SpinnakerC.h, [369](#)
- spinDeviceEventGetPayloadData
 - SpinnakerC.h, [369](#)
- spinDeviceEventGetPayloadDataSize
 - SpinnakerC.h, [370](#)
- spinDeviceEventHandler
 - SpinnakerDefsC.h, [453](#)
- spinDeviceEventHandlerCreate
 - SpinnakerC.h, [370](#)
- spinDeviceEventHandlerDestroy
 - SpinnakerC.h, [371](#)
- spinDeviceIndicatorModeEnums
 - Camera Enumerations, [82](#)
- spinDeviceLinkHeartbeatModeEnums
 - Camera Enumerations, [82](#)
- spinDeviceLinkThroughputLimitModeEnums
 - Camera Enumerations, [82](#)
- spinDevicePowerSupplySelectorEnums
 - Camera Enumerations, [82](#)
- spinDeviceRegistersEndiannessEnums
 - Camera Enumerations, [83](#)
- spinDeviceRemovalEventHandler
 - SpinnakerDefsC.h, [453](#)
- spinDeviceRemovalEventHandlerCreate
 - SpinnakerC.h, [371](#)
- spinDeviceRemovalEventHandlerDestroy
 - SpinnakerC.h, [372](#)
- spinDeviceScanTypeEnums
 - Camera Enumerations, [83](#)
- spinDeviceSerialPortBaudRateEnums
 - Camera Enumerations, [83](#)
- spinDeviceSerialPortSelectorEnums
 - Camera Enumerations, [84](#)
- spinDeviceStreamChannelEndiannessEnums
 - Camera Enumerations, [84](#)
- spinDeviceStreamChannelTypeEnums
 - Camera Enumerations, [84](#)
- spinDeviceTapGeometryEnums
 - Camera Enumerations, [85](#)
- spinDeviceTemperatureSelectorEnums
 - Camera Enumerations, [86](#)
- spinDeviceTLTypeEnums
 - Camera Enumerations, [86](#)
- spinDeviceTypeEnums
 - Camera Enumerations, [87](#)

- spinDisplayNotation
 - SpinnakerGenApiDefsC.h, [507](#)
- spinEncoderModeEnums
 - Camera Enumerations, [87](#)
- spinEncoderOutputModeEnums
 - Camera Enumerations, [87](#)
- spinEncoderResetActivationEnums
 - Camera Enumerations, [88](#)
- spinEncoderResetSourceEnums
 - Camera Enumerations, [88](#)
- spinEncoderSelectorEnums
 - Camera Enumerations, [89](#)
- spinEncoderSourceAEnums
 - Camera Enumerations, [90](#)
- spinEncoderSourceBEnums
 - Camera Enumerations, [90](#)
- spinEncoderStatusEnums
 - Camera Enumerations, [90](#)
- spinEndianess
 - SpinnakerGenApiDefsC.h, [507](#)
- spinEnumerationEntryGetEnumValue
 - SpinnakerGenApiC.h, [470](#)
- spinEnumerationEntryGetIntValue
 - SpinnakerGenApiC.h, [470](#)
- spinEnumerationEntryGetSymbolic
 - SpinnakerGenApiC.h, [471](#)
- spinEnumerationGetCurrentEntry
 - SpinnakerGenApiC.h, [471](#)
- spinEnumerationGetEntryByIndex
 - SpinnakerGenApiC.h, [472](#)
- spinEnumerationGetEntryByName
 - SpinnakerGenApiC.h, [472](#)
- spinEnumerationGetNumEntries
 - SpinnakerGenApiC.h, [473](#)
- spinEnumerationReleaseNode
 - SpinnakerGenApiC.h, [473](#)
- spinEnumerationSetEnumValue
 - SpinnakerGenApiC.h, [474](#)
- spinEnumerationSetIntValue
 - SpinnakerGenApiC.h, [474](#)
- spinError
 - SpinnakerDefsC.h, [457](#)
- spinErrorGetLast
 - SpinnakerC.h, [372](#)
- spinErrorGetLastBuildDate
 - SpinnakerC.h, [373](#)
- spinErrorGetLastBuildTime
 - SpinnakerC.h, [373](#)
- spinErrorGetLastFileName
 - SpinnakerC.h, [374](#)
- spinErrorGetLastFullMessage
 - SpinnakerC.h, [374](#)
- spinErrorGetLastFunctionName
 - SpinnakerC.h, [375](#)
- spinErrorGetLastLineNumber
 - SpinnakerC.h, [375](#)
- spinErrorGetLastMessage
 - SpinnakerC.h, [376](#)
- spinEventNotificationEnums
 - Camera Enumerations, [91](#)
- spinEventSelectorEnums
 - Camera Enumerations, [91](#)
- spinExposureActiveModeEnums
 - Camera Enumerations, [91](#)
- spinExposureAutoEnums
 - Camera Enumerations, [91](#)
- spinExposureModeEnums
 - Camera Enumerations, [92](#)
- spinExposureTimeModeEnums
 - Camera Enumerations, [92](#)
- spinExposureTimeSelectorEnums
 - Camera Enumerations, [93](#)
- spinFileOpenModeEnums
 - Camera Enumerations, [93](#)
- spinFileOperationSelectorEnums
 - Camera Enumerations, [93](#)
- spinFileOperationStatusEnums
 - Camera Enumerations, [94](#)
- spinFileSelectorEnums
 - Camera Enumerations, [94](#)
- spinFloatGetMax
 - SpinnakerGenApiC.h, [475](#)
- spinFloatGetMin
 - SpinnakerGenApiC.h, [475](#)
- spinFloatGetRepresentation
 - SpinnakerGenApiC.h, [476](#)
- spinFloatGetUnit
 - SpinnakerGenApiC.h, [476](#)
- spinFloatGetValue
 - SpinnakerGenApiC.h, [477](#)
- spinFloatGetValueEx
 - SpinnakerGenApiC.h, [477](#)
- spinFloatSetValue
 - SpinnakerGenApiC.h, [478](#)
- spinFloatSetValueEx
 - SpinnakerGenApiC.h, [478](#)
- spinGainAutoBalanceEnums
 - Camera Enumerations, [94](#)
- spinGainAutoEnums
 - Camera Enumerations, [96](#)
- spinGainSelectorEnums
 - Camera Enumerations, [96](#)
- spinGevCCPEnums
 - Camera Enumerations, [96](#)
- spinGevCurrentPhysicalLinkConfigurationEnums
 - Camera Enumerations, [97](#)
- spinGevGVCPExtendedStatusCodesSelectorEnums
 - Camera Enumerations, [97](#)
- spinGevGVSPExtendedIDModeEnums
 - Camera Enumerations, [97](#)
- spinGevIEEE1588ClockAccuracyEnums
 - Camera Enumerations, [98](#)
- spinGevIEEE1588ModeEnums
 - Camera Enumerations, [98](#)
- spinGevIEEE1588StatusEnums
 - Camera Enumerations, [98](#)

- spinGevIPConfigurationStatusEnums
 - Camera Enumerations, [99](#)
- spinGevPhysicalLinkConfigurationEnums
 - Camera Enumerations, [99](#)
- spinGevSupportedOptionSelectorEnums
 - Camera Enumerations, [99](#)
- spinH264Option, [292](#)
 - bitrate, [292](#)
 - frameRate, [293](#)
 - height, [293](#)
 - reserved, [293](#)
 - width, [293](#)
- spinImage
 - SpinnakerDefsC.h, [453](#)
- spinImageCalculateStatistics
 - SpinnakerC.h, [376](#)
- spinImageCheckCRC
 - SpinnakerC.h, [377](#)
- spinImageChunkDataGetFloatValue
 - SpinnakerC.h, [377](#)
- spinImageChunkDataGetIntValue
 - SpinnakerC.h, [377](#)
- spinImageComponentSelectorEnums
 - Camera Enumerations, [100](#)
- spinImageCompressionJPEGFormatOptionEnums
 - Camera Enumerations, [101](#)
- spinImageCompressionModeEnums
 - Camera Enumerations, [101](#)
- spinImageCompressionRateOptionEnums
 - Camera Enumerations, [102](#)
- spinImageCreate
 - SpinnakerC.h, [377](#)
- spinImageCreateEmpty
 - SpinnakerC.h, [378](#)
- spinImageCreateEx
 - SpinnakerC.h, [378](#)
- spinImageCreateEx2
 - SpinnakerC.h, [379](#)
- spinImageDeepCopy
 - SpinnakerC.h, [380](#)
- spinImageDestroy
 - SpinnakerC.h, [380](#)
- spinImageEventFunction
 - SpinnakerDefsC.h, [453](#)
- spinImageEventHandler
 - SpinnakerDefsC.h, [453](#)
- spinImageEventHandlerCreate
 - SpinnakerC.h, [381](#)
- spinImageEventHandlerDestroy
 - SpinnakerC.h, [381](#)
- spinImageFileFormat
 - SpinnakerDefsC.h, [458](#)
- spinImageGetBitsPerPixel
 - SpinnakerC.h, [382](#)
- spinImageGetBufferSize
 - SpinnakerC.h, [382](#)
- spinImageGetChunkLayoutID
 - SpinnakerC.h, [383](#)
- spinImageGetColorProcessing
 - SpinnakerC.h, [383](#)
- spinImageGetData
 - SpinnakerC.h, [384](#)
- spinImageGetFrameID
 - SpinnakerC.h, [384](#)
- spinImageGetHeight
 - SpinnakerC.h, [385](#)
- spinImageGetID
 - SpinnakerC.h, [385](#)
- spinImageGetOffsetX
 - SpinnakerC.h, [386](#)
- spinImageGetOffsetY
 - SpinnakerC.h, [386](#)
- spinImageGetPaddingX
 - SpinnakerC.h, [387](#)
- spinImageGetPaddingY
 - SpinnakerC.h, [387](#)
- spinImageGetPayloadType
 - SpinnakerC.h, [388](#)
- spinImageGetPixelFormat
 - SpinnakerC.h, [388](#)
- spinImageGetPixelFormatName
 - SpinnakerC.h, [389](#)
- spinImageGetPrivateData
 - SpinnakerC.h, [389](#)
- spinImageGetSize
 - SpinnakerC.h, [390](#)
- spinImageGetStatus
 - SpinnakerC.h, [390](#)
- spinImageGetStatusDescription
 - SpinnakerC.h, [391](#)
- spinImageGetStride
 - SpinnakerC.h, [391](#)
- spinImageGetTimeStamp
 - SpinnakerC.h, [392](#)
- spinImageGetTLPayloadType
 - SpinnakerC.h, [392](#)
- spinImageGetTLPixelFormat
 - SpinnakerC.h, [393](#)
- spinImageGetTLPixelFormatNamespace
 - SpinnakerC.h, [393](#)
- spinImageGetValidPayloadSize
 - SpinnakerC.h, [394](#)
- spinImageGetWidth
 - SpinnakerC.h, [394](#)
- spinImageHasCRC
 - SpinnakerC.h, [395](#)
- spinImageIsIncomplete
 - SpinnakerC.h, [395](#)
- spinImageList
 - SpinnakerDefsC.h, [453](#)
- spinImageListAppend
 - SpinnakerC.h, [396](#)
- spinImageListClear
 - SpinnakerC.h, [396](#)
- spinImageListCreateEmpty
 - SpinnakerC.h, [396](#)

- spinImageListDestroy
 - SpinnakerC.h, [397](#)
- spinImageListEventFunction
 - SpinnakerDefsC.h, [454](#)
- spinImageListEventHandler
 - SpinnakerDefsC.h, [454](#)
- spinImageListEventHandlerCreate
 - SpinnakerC.h, [397](#)
- spinImageListEventHandlerDestroy
 - SpinnakerC.h, [398](#)
- spinImageListGet
 - SpinnakerC.h, [398](#)
- spinImageListGetByPixelFormat
 - SpinnakerC.h, [399](#)
- spinImageListGetSize
 - SpinnakerC.h, [399](#)
- spinImageListLoad
 - SpinnakerC.h, [400](#)
- spinImageListRelease
 - SpinnakerC.h, [400](#)
- spinImageListRemove
 - SpinnakerC.h, [400](#)
- spinImageListRemoveByPixelFormat
 - SpinnakerC.h, [401](#)
- spinImageListSave
 - SpinnakerC.h, [401](#)
- spinImageProcessor
 - SpinnakerDefsC.h, [454](#)
- spinImageProcessorApplyGamma
 - SpinnakerC.h, [402](#)
- spinImageProcessorConvert
 - SpinnakerC.h, [402](#)
- spinImageProcessorConvertImageList
 - SpinnakerC.h, [403](#)
- spinImageProcessorCreate
 - SpinnakerC.h, [404](#)
- spinImageProcessorDestroy
 - SpinnakerC.h, [405](#)
- spinImageProcessorGetColorProcessing
 - SpinnakerC.h, [405](#)
- spinImageProcessorGetNumDecompressionThreads
 - SpinnakerC.h, [405](#)
- spinImageProcessorSetColorProcessing
 - SpinnakerC.h, [406](#)
- spinImageProcessorSetNumDecompressionThreads
 - SpinnakerC.h, [406](#)
- spinImageRelease
 - SpinnakerC.h, [407](#)
- spinImageReset
 - SpinnakerC.h, [407](#)
- spinImageResetEx
 - SpinnakerC.h, [408](#)
- spinImageSave
 - SpinnakerC.h, [408](#)
- spinImageSaveBmp
 - SpinnakerC.h, [409](#)
- spinImageSaveFromExt
 - SpinnakerC.h, [409](#)
- spinImageSaveJpeg
 - SpinnakerC.h, [410](#)
- spinImageSaveJpg2
 - SpinnakerC.h, [410](#)
- spinImageSavePgm
 - SpinnakerC.h, [411](#)
- spinImageSavePng
 - SpinnakerC.h, [411](#)
- spinImageSavePpm
 - SpinnakerC.h, [412](#)
- spinImageSaveTiff
 - SpinnakerC.h, [412](#)
- spinImageStatistics
 - SpinnakerDefsC.h, [454](#)
- spinImageStatisticsCreate
 - SpinnakerC.h, [413](#)
- spinImageStatisticsDestroy
 - SpinnakerC.h, [413](#)
- spinImageStatisticsDisableAll
 - SpinnakerC.h, [414](#)
- spinImageStatisticsEnableAll
 - SpinnakerC.h, [414](#)
- spinImageStatisticsEnableGreyOnly
 - SpinnakerC.h, [415](#)
- spinImageStatisticsEnableHslOnly
 - SpinnakerC.h, [415](#)
- spinImageStatisticsEnableRgbOnly
 - SpinnakerC.h, [415](#)
- spinImageStatisticsGetAll
 - SpinnakerC.h, [416](#)
- spinImageStatisticsGetChannelStatus
 - SpinnakerC.h, [417](#)
- spinImageStatisticsGetHistogram
 - SpinnakerC.h, [417](#)
- spinImageStatisticsGetMean
 - SpinnakerC.h, [418](#)
- spinImageStatisticsGetNumPixelValues
 - SpinnakerC.h, [418](#)
- spinImageStatisticsGetPixelValueRange
 - SpinnakerC.h, [419](#)
- spinImageStatisticsGetRange
 - SpinnakerC.h, [419](#)
- spinImageStatisticsSetChannelStatus
 - SpinnakerC.h, [420](#)
- spinImageStatus
 - SpinnakerDefsC.h, [459](#)
- spinIncMode
 - SpinnakerGenApiDefsC.h, [508](#)
- spinInputDirection
 - SpinnakerGenApiDefsC.h, [508](#)
- spinIntegerGetInc
 - SpinnakerGenApiC.h, [479](#)
- spinIntegerGetMax
 - SpinnakerGenApiC.h, [479](#)
- spinIntegerGetMin
 - SpinnakerGenApiC.h, [480](#)
- spinIntegerGetRepresentation
 - SpinnakerGenApiC.h, [480](#)

- spinIntegerGetValue
 - SpinnakerGenApiC.h, [481](#)
- spinIntegerGetValueEx
 - SpinnakerGenApiC.h, [481](#)
- spinIntegerSetValue
 - SpinnakerGenApiC.h, [482](#)
- spinIntegerSetValueEx
 - SpinnakerGenApiC.h, [482](#)
- spinInterface
 - SpinnakerDefsC.h, [454](#)
- spinInterfaceEventHandler
 - SpinnakerDefsC.h, [454](#)
- spinInterfaceEventHandlerCreate
 - SpinnakerC.h, [420](#)
- spinInterfaceEventHandlerDestroy
 - SpinnakerC.h, [421](#)
- spinInterfaceGetCameras
 - SpinnakerC.h, [421](#)
- spinInterfaceGetCamerasEx
 - SpinnakerC.h, [422](#)
- spinInterfaceGetTLNodeMap
 - SpinnakerC.h, [422](#)
- spinInterfaceIsInUse
 - SpinnakerC.h, [423](#)
- spinInterfaceList
 - SpinnakerDefsC.h, [455](#)
- spinInterfaceListClear
 - SpinnakerC.h, [423](#)
- spinInterfaceListCreateEmpty
 - SpinnakerC.h, [423](#)
- spinInterfaceListDestroy
 - SpinnakerC.h, [424](#)
- spinInterfaceListGet
 - SpinnakerC.h, [424](#)
- spinInterfaceListGetSize
 - SpinnakerC.h, [425](#)
- spinInterfaceRegisterDeviceArrivalEventHandler
 - SpinnakerC.h, [425](#)
- spinInterfaceRegisterDeviceRemovalEventHandler
 - SpinnakerC.h, [426](#)
- spinInterfaceRegisterInterfaceEventHandler
 - SpinnakerC.h, [426](#)
- spinInterfaceRelease
 - SpinnakerC.h, [427](#)
- spinInterfaceSendActionCommand
 - SpinnakerC.h, [427](#)
- spinInterfaceType
 - SpinnakerGenApiDefsC.h, [508](#)
- spinInterfaceUnregisterDeviceArrivalEventHandler
 - SpinnakerC.h, [428](#)
- spinInterfaceUnregisterDeviceRemovalEventHandler
 - SpinnakerC.h, [428](#)
- spinInterfaceUnregisterInterfaceEventHandler
 - SpinnakerC.h, [429](#)
- spinInterfaceUpdateCameras
 - SpinnakerC.h, [429](#)
- spinJPEGOption, [293](#)
 - progressive, [294](#)
 - quality, [294](#)
 - reserved, [294](#)
- spinJPG2Option, [295](#)
 - quality, [295](#)
 - reserved, [295](#)
- spinLibraryVersion, [295](#)
 - build, [296](#)
 - major, [296](#)
 - minor, [296](#)
 - type, [296](#)
- spinLineFormatEnums
 - Camera Enumerations, [102](#)
- spinLineInputFilterSelectorEnums
 - Camera Enumerations, [102](#)
- spinLineModeEnums
 - Camera Enumerations, [103](#)
- spinLineSelectorEnums
 - Camera Enumerations, [103](#)
- spinLineSourceEnums
 - Camera Enumerations, [103](#)
- spinLinkType
 - SpinnakerGenApiDefsC.h, [509](#)
- spinLogDataGetCategoryName
 - SpinnakerC.h, [430](#)
- spinLogDataGetLogMessage
 - SpinnakerC.h, [430](#)
- spinLogDataGetNDC
 - SpinnakerC.h, [431](#)
- spinLogDataGetPriority
 - SpinnakerC.h, [431](#)
- spinLogDataGetPriorityName
 - SpinnakerC.h, [432](#)
- spinLogDataGetThreadName
 - SpinnakerC.h, [432](#)
- spinLogDataGetTimestamp
 - SpinnakerC.h, [433](#)
- spinLogEventData
 - SpinnakerDefsC.h, [455](#)
- spinLogEventFunction
 - SpinnakerDefsC.h, [455](#)
- spinLogEventHandler
 - SpinnakerDefsC.h, [455](#)
- spinLogEventHandlerCreate
 - SpinnakerC.h, [433](#)
- spinLogEventHandlerDestroy
 - SpinnakerC.h, [434](#)
- spinLogicBlockLUTInputActivationEnums
 - Camera Enumerations, [104](#)
- spinLogicBlockLUTInputSelectorEnums
 - Camera Enumerations, [104](#)
- spinLogicBlockLUTInputSourceEnums
 - Camera Enumerations, [105](#)
- spinLogicBlockLUTSelectorEnums
 - Camera Enumerations, [105](#)
- spinLogicBlockSelectorEnums
 - Camera Enumerations, [106](#)
- spinLUTSelectorEnums
 - Camera Enumerations, [106](#)

- spinMJPGOption, 297
 - frameRate, 297
 - height, 297
 - quality, 298
 - reserved, 298
 - width, 298
- Spinnaker C API, 142
- Spinnaker C Definitions, 23
- Spinnaker C Enumerations, 148
- Spinnaker C Function Signatures, 148
- Spinnaker C GenICam API, 148
- Spinnaker C GenICam Enumerations, 151
- Spinnaker C GenICam Handles, 150
- Spinnaker C Handles, 148
- Spinnaker C QuickSpin API, 141
- Spinnaker C Structures, 148
- SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LASER
 - SpinnakerDefsC.h, 456
- SPINNAKER_ACTION_COMMAND_STATUS_ERROR
 - SpinnakerDefsC.h, 456
- SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TONE
 - SpinnakerDefsC.h, 456
- SPINNAKER_ACTION_COMMAND_STATUS_OK
 - SpinnakerDefsC.h, 456
- SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW
 - SpinnakerDefsC.h, 456
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILinear
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECTIONAL_FILTER
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSING
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQSPINNAKER
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPPSPINNAKER
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR_INVALID
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NONSPINNAKER
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGIDSPINNAKER
 - SpinnakerDefsC.h, 457
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEIGHTED_DIRECTIONAL_FILTER
 - SpinnakerDefsC.h, 457
- SPINNAKER_ERR_ABORT
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_ACCESS_DENIED
 - SpinnakerDefsC.h, 457
- SPINNAKER_ERR_BUFFER_TOO_SMALL
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_BUSY
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_CUSTOM_ID
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_ERROR
 - SpinnakerDefsC.h, 457
- SPINNAKER_ERR_GENICAM_ACCESS
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_BAD_ALLOCATION
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_DYNAMIC_CAST
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_GENERIC
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_LOGICAL
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_OUT_OF_RANGE
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_PROPERTY
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_RUN_TIME
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_GENICAM_TIMEOUT
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IM_COLOR_CONVERSION
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IM_CONVERT
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IM_COPY
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IM_HISTOGRAM_MEAN
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IM_HISTOGRAM_RANGE
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IM_MALLOC
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IM_MIN_MAX
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IM_NOT_SUPPORTED
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_INVALID_ADDRESS
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_INVALID_BUFFER
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_INVALID_HANDLE
 - SpinnakerDefsC.h, 457
- SPINNAKER_ERR_INVALID_ID
 - SpinnakerDefsC.h, 457
- SPINNAKER_ERR_INVALID_INDEX
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_INVALID_PARAMETER
 - SpinnakerDefsC.h, 457
- SPINNAKER_ERR_INVALID_VALUE
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_IO
 - SpinnakerDefsC.h, 457
- SPINNAKER_ERR_NO_DATA
 - SpinnakerDefsC.h, 457
- SPINNAKER_ERR_NOT_AVAILABLE
 - SpinnakerDefsC.h, 458
- SPINNAKER_ERR_NOT_IMPLEMENTED
 - SpinnakerDefsC.h, 457

- SPINNAKER_ERR_NOT_INITIALIZED
SpinnakerDefsC.h, [457](#)
- SPINNAKER_ERR_OUT_OF_MEMORY
SpinnakerDefsC.h, [458](#)
- SPINNAKER_ERR_PARSING_CHUNK_DATA
SpinnakerDefsC.h, [458](#)
- SPINNAKER_ERR_RESOURCE_EXHAUSTED
SpinnakerDefsC.h, [458](#)
- SPINNAKER_ERR_RESOURCE_IN_USE
SpinnakerDefsC.h, [457](#)
- SPINNAKER_ERR_SUCCESS
SpinnakerDefsC.h, [457](#)
- SPINNAKER_ERR_TIMEOUT
SpinnakerDefsC.h, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_BMP
SpinnakerDefsC.h, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT
SpinnakerDefsC.h, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PGM
SpinnakerDefsC.h, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PNG
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PPM
SpinnakerDefsC.h, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_RAW
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_TIFF
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_MISSING_LEADER
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_MISSING_PACKETS
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_MISSING_TRAILER
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_NO_ERROR
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT
SpinnakerDefsC.h, [459](#)
- SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR
SpinnakerDefsC.h, [459](#)
- SPINNAKER_LOG_LEVEL_ALERT
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_CRIT
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_DEBUG
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_ERROR
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_FATAL
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_INFO
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_NOTICE
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_NOTSET
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_OFF
SpinnakerDefsC.h, [460](#)
- SPINNAKER_LOG_LEVEL_WARN
SpinnakerDefsC.h, [460](#)
- SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID
SpinnakerDefsC.h, [462](#)
- SPINNAKER_STATISTICS_CHANNEL_BLUE
SpinnakerDefsC.h, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_GREEN
SpinnakerDefsC.h, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_GREY
SpinnakerDefsC.h, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_HUE
SpinnakerDefsC.h, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS
SpinnakerDefsC.h, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS
SpinnakerDefsC.h, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_RED
SpinnakerDefsC.h, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_SATURATION
SpinnakerDefsC.h, [460](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE
SpinnakerDefsC.h, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3
SpinnakerDefsC.h, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4
SpinnakerDefsC.h, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE
SpinnakerDefsC.h, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_JPG
SpinnakerDefsC.h, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_LZW
SpinnakerDefsC.h, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_NONE
SpinnakerDefsC.h, [460](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS
SpinnakerDefsC.h, [461](#)

- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_FILE
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_H264
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_IMAGE
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG2000
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN
 - SpinnakerDefsC.h, [461](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV
 - SpinnakerDefsC.h, [462](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC
 - SpinnakerDefsC.h, [462](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT
 - SpinnakerDefsC.h, [462](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT
 - SpinnakerDefsC.h, [462](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN
 - SpinnakerDefsC.h, [462](#)
- SpinnakerC.h
 - pblsStreaming, [446](#)
 - spinCameraBeginAcquisition, [351](#)
 - spinCameraDelInit, [351](#)
 - spinCameraDiscoverMaxPacketSize, [351](#)
 - spinCameraEndAcquisition, [352](#)
 - spinCameraForceIP, [352](#)
 - spinCameraGetAccessMode, [353](#)
 - spinCameraGetDeviceID, [353](#)
 - spinCameraGetGuiXml, [354](#)
 - spinCameraGetNextImage, [354](#)
 - spinCameraGetNextImageEx, [355](#)
 - spinCameraGetNextImageSync, [355](#)
 - spinCameraGetNodeMap, [356](#)
 - spinCameraGetTLDeviceNodeMap, [356](#)
 - spinCameraGetTLStreamNodeMap, [357](#)
 - spinCameraInit, [357](#)
 - spinCamerasInitialized, [358](#)
 - spinCamerasValid, [358](#)
 - spinCameraListAppend, [359](#)
 - spinCameraListClear, [359](#)
 - spinCameraListCreateEmpty, [359](#)
 - spinCameraListDestroy, [360](#)
 - spinCameraListGet, [360](#)
 - spinCameraListGetBySerial, [361](#)
 - spinCameraListGetSize, [361](#)
 - spinCameraListRemove, [362](#)
 - spinCameraListRemoveBySerial, [362](#)
 - spinCameraReadPort, [363](#)
 - spinCameraRegisterDeviceEventHandler, [363](#)
 - spinCameraRegisterDeviceEventHandlerEx, [363](#)
 - spinCameraRegisterImageEventHandler, [364](#)
 - spinCameraRegisterImageEventHandlerEx, [364](#)
 - spinCameraRegisterImageListEventHandler, [365](#)
 - spinCameraRelease, [365](#)
 - spinCameraUnregisterDeviceEventHandler, [366](#)
 - spinCameraUnregisterImageEventHandler, [366](#)
 - spinCameraUnregisterImageListEventHandler, [367](#)
 - spinCameraWritePort, [367](#)
 - spinDeviceArrivalEventHandlerCreate, [367](#)
 - spinDeviceArrivalEventHandlerDestroy, [368](#)
 - spinDeviceEventGetId, [368](#)
 - spinDeviceEventGetName, [369](#)
 - spinDeviceEventGetPayloadData, [369](#)
 - spinDeviceEventGetPayloadDataSize, [370](#)
 - spinDeviceEventHandlerCreate, [370](#)
 - spinDeviceEventHandlerDestroy, [371](#)
 - spinDeviceRemovalEventHandlerCreate, [371](#)
 - spinDeviceRemovalEventHandlerDestroy, [372](#)
 - spinErrorGetLast, [372](#)
 - spinErrorGetLastBuildDate, [373](#)
 - spinErrorGetLastBuildTime, [373](#)
 - spinErrorGetLastFileName, [374](#)
 - spinErrorGetLastFullMessage, [374](#)
 - spinErrorGetLastFunctionName, [375](#)
 - spinErrorGetLastLineNumber, [375](#)
 - spinErrorGetLastMessage, [376](#)
 - spinImageCalculateStatistics, [376](#)
 - spinImageCheckCRC, [377](#)
 - spinImageChunkDataGetFloatValue, [377](#)
 - spinImageChunkDataGetIntValue, [377](#)
 - spinImageCreate, [377](#)
 - spinImageCreateEmpty, [378](#)
 - spinImageCreateEx, [378](#)
 - spinImageCreateEx2, [379](#)
 - spinImageDeepCopy, [380](#)
 - spinImageDestroy, [380](#)
 - spinImageEventHandlerCreate, [381](#)
 - spinImageEventHandlerDestroy, [381](#)
 - spinImageGetBitsPerPixel, [382](#)
 - spinImageGetBufferSize, [382](#)
 - spinImageGetChunkLayoutID, [383](#)
 - spinImageGetColorProcessing, [383](#)
 - spinImageGetData, [384](#)
 - spinImageGetFrameID, [384](#)
 - spinImageGetHeight, [385](#)

- spinImageGetID, 385
- spinImageGetOffsetX, 386
- spinImageGetOffsetY, 386
- spinImageGetPaddingX, 387
- spinImageGetPaddingY, 387
- spinImageGetPayloadType, 388
- spinImageGetPixelFormat, 388
- spinImageGetPixelFormatName, 389
- spinImageGetPrivateData, 389
- spinImageGetSize, 390
- spinImageGetStatus, 390
- spinImageGetStatusDescription, 391
- spinImageGetStride, 391
- spinImageGetTimeStamp, 392
- spinImageGetTLPayloadType, 392
- spinImageGetTLPixelFormat, 393
- spinImageGetTLPixelFormatNamespace, 393
- spinImageGetValidPayloadSize, 394
- spinImageGetWidth, 394
- spinImageHasCRC, 395
- spinImageIsIncomplete, 395
- spinImageListAppend, 396
- spinImageListClear, 396
- spinImageListCreateEmpty, 396
- spinImageListDestroy, 397
- spinImageListEventHandlerCreate, 397
- spinImageListEventHandlerDestroy, 398
- spinImageListGet, 398
- spinImageListGetByPixelFormat, 399
- spinImageListGetSize, 399
- spinImageListLoad, 400
- spinImageListRelease, 400
- spinImageListRemove, 400
- spinImageListRemoveByPixelFormat, 401
- spinImageListSave, 401
- spinImageProcessorApplyGamma, 402
- spinImageProcessorConvert, 402
- spinImageProcessorConvertImageList, 403
- spinImageProcessorCreate, 404
- spinImageProcessorDestroy, 405
- spinImageProcessorGetColorProcessing, 405
- spinImageProcessorGetNumDecompression-
Threads, 405
- spinImageProcessorSetColorProcessing, 406
- spinImageProcessorSetNumDecompression-
Threads, 406
- spinImageRelease, 407
- spinImageReset, 407
- spinImageResetEx, 408
- spinImageSave, 408
- spinImageSaveBmp, 409
- spinImageSaveFromExt, 409
- spinImageSaveJpeg, 410
- spinImageSaveJpg2, 410
- spinImageSavePgm, 411
- spinImageSavePng, 411
- spinImageSavePpm, 412
- spinImageSaveTiff, 412
- spinImageStatisticsCreate, 413
- spinImageStatisticsDestroy, 413
- spinImageStatisticsDisableAll, 414
- spinImageStatisticsEnableAll, 414
- spinImageStatisticsEnableGreyOnly, 415
- spinImageStatisticsEnableHslOnly, 415
- spinImageStatisticsEnableRgbOnly, 415
- spinImageStatisticsGetAll, 416
- spinImageStatisticsGetChannelStatus, 417
- spinImageStatisticsGetHistogram, 417
- spinImageStatisticsGetMean, 418
- spinImageStatisticsGetNumPixelValues, 418
- spinImageStatisticsGetPixelValueRange, 419
- spinImageStatisticsGetRange, 419
- spinImageStatisticsSetChannelStatus, 420
- spinInterfaceEventHandlerCreate, 420
- spinInterfaceEventHandlerDestroy, 421
- spinInterfaceGetCameras, 421
- spinInterfaceGetCamerasEx, 422
- spinInterfaceGetTLNodeMap, 422
- spinInterfaceIsInUse, 423
- spinInterfaceListClear, 423
- spinInterfaceListCreateEmpty, 423
- spinInterfaceListDestroy, 424
- spinInterfaceListGet, 424
- spinInterfaceListGetSize, 425
- spinInterfaceRegisterDeviceArrivalEventHandler, 425
- spinInterfaceRegisterDeviceRemovalEventHandler, 426
- spinInterfaceRegisterInterfaceEventHandler, 426
- spinInterfaceRelease, 427
- spinInterfaceSendActionCommand, 427
- spinInterfaceUnregisterDeviceArrivalEventHandler, 428
- spinInterfaceUnregisterDeviceRemovalEventHandler, 428
- spinInterfaceUnregisterInterfaceEventHandler, 429
- spinInterfaceUpdateCameras, 429
- spinLogDataGetCategoryName, 430
- spinLogDataGetLogMessage, 430
- spinLogDataGetNDC, 431
- spinLogDataGetPriority, 431
- spinLogDataGetPriorityName, 432
- spinLogDataGetThreadName, 432
- spinLogDataGetTimestamp, 433
- spinLogEventHandlerCreate, 433
- spinLogEventHandlerDestroy, 434
- SPINNAKERC_API_DEPRECATED, 434
- spinSystemGetCameras, 435
- spinSystemGetCamerasEx, 436
- spinSystemGetInstance, 436
- spinSystemGetInterfaces, 437
- spinSystemGetLibraryVersion, 437
- spinSystemGetLoggingLevel, 437
- spinSystemGetTLNodeMap, 438
- spinSystemIsInUse, 438

- spinSystemRegisterDeviceArrivalEventHandler, [439](#)
- spinSystemRegisterDeviceRemovalEventHandler, [439](#)
- spinSystemRegisterInterfaceEventHandler, [440](#)
- spinSystemRegisterLogEventHandler, [440](#)
- spinSystemReleaseInstance, [441](#)
- spinSystemSendActionCommand, [441](#)
- spinSystemSetLoggingLevel, [442](#)
- spinSystemUnregisterAllLogEventHandlers, [443](#)
- spinSystemUnregisterDeviceArrivalEventHandler, [443](#)
- spinSystemUnregisterDeviceRemovalEventHandler, [444](#)
- spinSystemUnregisterInterfaceEventHandler, [444](#)
- spinSystemUnregisterLogEventHandler, [445](#)
- spinSystemUpdateCameras, [445](#)
- spinSystemUpdateCamerasEx, [446](#)
- SPINNAKERC_API
 - SpinnakerPlatformC.h, [514](#)
- SPINNAKERC_API_DEPRECATED
 - SpinnakerC.h, [434](#)
- SpinnakerDefsC.h
 - bool8_t, [452](#)
 - False, [462](#)
 - spinActionCommandStatus, [456](#)
 - spinArrivalEventFunction, [452](#)
 - spinCamera, [452](#)
 - spinCameraList, [452](#)
 - spinColorProcessingAlgorithm, [456](#)
 - spinDeviceArrivalEventHandler, [452](#)
 - spinDeviceEventData, [452](#)
 - spinDeviceEventFunction, [452](#)
 - spinDeviceEventHandler, [453](#)
 - spinDeviceRemovalEventHandler, [453](#)
 - spinError, [457](#)
 - spinImage, [453](#)
 - spinImageEventFunction, [453](#)
 - spinImageEventHandler, [453](#)
 - spinImageFileFormat, [458](#)
 - spinImageList, [453](#)
 - spinImageListEventFunction, [454](#)
 - spinImageListEventHandler, [454](#)
 - spinImageProcessor, [454](#)
 - spinImageStatistics, [454](#)
 - spinImageStatus, [459](#)
 - spinInterface, [454](#)
 - spinInterfaceEventHandler, [454](#)
 - spinInterfaceList, [455](#)
 - spinLogEventData, [455](#)
 - spinLogEventFunction, [455](#)
 - spinLogEventHandler, [455](#)
 - SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE, [456](#)
 - SPINNAKER_ACTION_COMMAND_STATUS_ERROR, [456](#)
 - SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TSN, [456](#)
 - SPINNAKER_ACTION_COMMAND_STATUS_OK, [456](#)
 - SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW, [456](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILINEAR, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECTIONAL, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSITIVE, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQ_LINEAR, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPP, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR_4, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_NONE, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGOROUS, [457](#)
 - SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEIGHTED_NEAREST_NEIGHBOR, [457](#)
 - SPINNAKER_ERR_ABORT, [458](#)
 - SPINNAKER_ERR_ACCESS_DENIED, [457](#)
 - SPINNAKER_ERR_BUFFER_TOO_SMALL, [458](#)
 - SPINNAKER_ERR_BUSY, [458](#)
 - SPINNAKER_ERR_CUSTOM_ID, [458](#)
 - SPINNAKER_ERR_ERROR, [457](#)
 - SPINNAKER_ERR_GENICAM_ACCESS, [458](#)
 - SPINNAKER_ERR_GENICAM_BAD_ALLOCATION, [458](#)
 - SPINNAKER_ERR_GENICAM_DYNAMIC_CAST, [458](#)
 - SPINNAKER_ERR_GENICAM_GENERIC, [458](#)
 - SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT, [458](#)
 - SPINNAKER_ERR_GENICAM_LOGICAL, [458](#)
 - SPINNAKER_ERR_GENICAM_OUT_OF_RANGE, [458](#)
 - SPINNAKER_ERR_GENICAM_PROPERTY, [458](#)
 - SPINNAKER_ERR_GENICAM_RUN_TIME, [458](#)
 - SPINNAKER_ERR_GENICAM_TIMEOUT, [458](#)
 - SPINNAKER_ERR_IM_COLOR_CONVERSION, [458](#)
 - SPINNAKER_ERR_IM_CONVERT, [458](#)
 - SPINNAKER_ERR_IM_COPY, [458](#)
 - SPINNAKER_ERR_IM_HISTOGRAM_MEAN, [458](#)
 - SPINNAKER_ERR_IM_HISTOGRAM_RANGE, [458](#)
 - SPINNAKER_ERR_IM_MALLOC, [458](#)
 - SPINNAKER_ERR_IM_MIN_MAX, [458](#)
 - SPINNAKER_ERR_IM_NOT_SUPPORTED, [458](#)
 - SPINNAKER_ERR_INVALID_ADDRESS, [458](#)
 - SPINNAKER_ERR_INVALID_BUFFER, [458](#)
 - SPINNAKER_ERR_INVALID_HANDLE, [457](#)
 - SPINNAKER_ERR_INVALID_ID, [457](#)

- SPINNAKER_ERR_INVALID_INDEX, [458](#)
- SPINNAKER_ERR_INVALID_PARAMETER, [457](#)
- SPINNAKER_ERR_INVALID_VALUE, [458](#)
- SPINNAKER_ERR_IO, [457](#)
- SPINNAKER_ERR_NO_DATA, [457](#)
- SPINNAKER_ERR_NOT_AVAILABLE, [458](#)
- SPINNAKER_ERR_NOT_IMPLEMENTED, [457](#)
- SPINNAKER_ERR_NOT_INITIALIZED, [457](#)
- SPINNAKER_ERR_OUT_OF_MEMORY, [458](#)
- SPINNAKER_ERR_PARSING_CHUNK_DATA, [458](#)
- SPINNAKER_ERR_RESOURCE_EXHAUSTED, [458](#)
- SPINNAKER_ERR_RESOURCE_IN_USE, [457](#)
- SPINNAKER_ERR_SUCCESS, [457](#)
- SPINNAKER_ERR_TIMEOUT, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_BMP, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PGM, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PNG, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PPM, [458](#)
- SPINNAKER_IMAGE_FILE_FORMAT_RAW, [459](#)
- SPINNAKER_IMAGE_FILE_FORMAT_TIFF, [459](#)
- SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID, [459](#)
- SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED, [459](#)
- SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE, [459](#)
- SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW, [459](#)
- SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT, [459](#)
- SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT, [459](#)
- SPINNAKER_IMAGE_STATUS_MISSING_LEADER, [459](#)
- SPINNAKER_IMAGE_STATUS_MISSING_PACKETS, [459](#)
- SPINNAKER_IMAGE_STATUS_MISSING_TRAILER, [459](#)
- SPINNAKER_IMAGE_STATUS_NO_ERROR, [459](#)
- SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES, [459](#)
- SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT, [459](#)
- SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT, [459](#)
- SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR, [459](#)
- SPINNAKER_LOG_LEVEL_ALERT, [460](#)
- SPINNAKER_LOG_LEVEL_CRIT, [460](#)
- SPINNAKER_LOG_LEVEL_DEBUG, [460](#)
- SPINNAKER_LOG_LEVEL_ERROR, [460](#)
- SPINNAKER_LOG_LEVEL_FATAL, [460](#)
- SPINNAKER_LOG_LEVEL_INFO, [460](#)
- SPINNAKER_LOG_LEVEL_NOTICE, [460](#)
- SPINNAKER_LOG_LEVEL_NOTSET, [460](#)
- SPINNAKER_LOG_LEVEL_OFF, [460](#)
- SPINNAKER_LOG_LEVEL_WARN, [460](#)
- SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID, [462](#)
- SPINNAKER_STATISTICS_CHANNEL_BLUE, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_GREEN, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_GREY, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_HUE, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS, [460](#)
- SPINNAKER_STATISTICS_CHANNEL_RED, [460](#)
- SPINNAKER_STATISTICS_CHANNEL SATURATION, [460](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_JPG, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_LZW, [461](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_NONE, [460](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_FILE, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_H264, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_IMAGE, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG2000, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESS, [461](#)
- SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED, [461](#)

- SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED, 461
- SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART, 461
- SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA, 461
- SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN, 461
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV, 462
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC, 462
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT, 462
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT, 462
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN, 462
- spinnakerLogLevel, 459
- spinRemovalEventFunction, 455
- spinStatisticsChannel, 460
- spinSystem, 455
- spinTIFFCompressionMethod, 460
- spinTLPayloadType, 461
- spinTLPixelFormatNamespace, 461
- spinVideo, 456
- True, 462
- SpinnakerGenApiC.h
 - spinBooleanGetValue, 466
 - spinBooleanSetValue, 467
 - spinCategoryGetFeatureByIndex, 467
 - spinCategoryGetNumFeatures, 468
 - spinCategoryReleaseNode, 468
 - spinCommandExecute, 469
 - spinCommandIsDone, 469
 - spinEnumerationEntryGetEnumValue, 470
 - spinEnumerationEntryGetIntValue, 470
 - spinEnumerationEntryGetSymbolic, 471
 - spinEnumerationGetCurrentEntry, 471
 - spinEnumerationGetEntryByIndex, 472
 - spinEnumerationGetEntryByName, 472
 - spinEnumerationGetNumEntries, 473
 - spinEnumerationReleaseNode, 473
 - spinEnumerationSetEnumValue, 474
 - spinEnumerationSetIntValue, 474
 - spinFloatGetMax, 475
 - spinFloatGetMin, 475
 - spinFloatGetRepresentation, 476
 - spinFloatGetUnit, 476
 - spinFloatGetValue, 477
 - spinFloatGetValueEx, 477
 - spinFloatSetValue, 478
 - spinFloatSetValueEx, 478
 - spinIntegerGetInc, 479
 - spinIntegerGetMax, 479
 - spinIntegerGetMin, 480
 - spinIntegerGetRepresentation, 480
 - spinIntegerGetValue, 481
 - spinIntegerGetValueEx, 481
 - spinIntegerSetValue, 482
 - spinIntegerSetValueEx, 482
 - spinNodeDeregisterCallback, 483
 - spinNodeFromString, 483
 - spinNodeFromStringEx, 484
 - spinNodeGetAccessMode, 484
 - spinNodeGetCachingMode, 485
 - spinNodeGetDescription, 485
 - spinNodeGetDisplayName, 486
 - spinNodeGetImposedAccessMode, 486
 - spinNodeGetImposedVisibility, 487
 - spinNodeGetName, 487
 - spinNodeGetNameSpace, 488
 - spinNodeGetPollingTime, 488
 - spinNodeGetToolTip, 489
 - spinNodeGetType, 489
 - spinNodeGetVisibility, 490
 - spinNodeInvalidateNode, 490
 - spinNodesAvailable, 491
 - spinNodesEqual, 491
 - spinNodesImplemented, 491
 - spinNodesReadable, 492
 - spinNodesWritable, 492
 - spinNodeMapGetNode, 493
 - spinNodeMapGetNodeByIndex, 493
 - spinNodeMapGetNumNodes, 494
 - spinNodeMapPoll, 494
 - spinNodeMapReleaseNode, 495
 - spinNodeRegisterCallback, 495
 - spinNodeToString, 496
 - spinNodeToStringEx, 496
 - spinRegisterGet, 497
 - spinRegisterGetAddress, 497
 - spinRegisterGetEx, 498
 - spinRegisterGetLength, 498
 - spinRegisterSet, 499
 - spinRegisterSetEx, 499
 - spinRegisterSetReference, 500
 - spinStringGetMaxLength, 500
 - spinStringGetValue, 501
 - spinStringGetValueEx, 501
 - spinStringSetValue, 502
 - spinStringSetValueEx, 502
- SpinnakerGenApiDefsC.h
 - _CycleDetectAccessMode, 507
 - _UndefinedAccessMode, 507
 - _UndefinedCachingMode, 507
 - _UndefinedEDisplayNotation, 507
 - _UndefinedESlope, 512
 - _UndefinedEXMLValidation, 513
 - _UndefinedEndian, 508
 - _UndefinedNameSpace, 510
 - _UndefinedRepresentation, 511
 - _UndefinedSign, 511
 - _UndefinedStandardNameSpace, 512
 - _UndefinedVisibility, 512
 - _UndefinedYesNo, 513

- Automatic, [512](#)
- BaseNode, [510](#)
- Beginner, [512](#)
- BigEndian, [508](#)
- Boolean, [511](#)
- BooleanNode, [510](#)
- CategoryNode, [511](#)
- CL, [512](#)
- CommandNode, [510](#)
- ctAllDependingNodes, [510](#)
- ctAllTerminalNodes, [510](#)
- ctDependingChildren, [510](#)
- ctInvalidators, [510](#)
- ctReadingChildren, [510](#)
- ctWritingChildren, [510](#)
- Custom, [510](#)
- Decreasing, [512](#)
- EnumEntryNode, [511](#)
- EnumerationNode, [511](#)
- Expert, [512](#)
- fixedIncrement, [508](#)
- FloatNode, [510](#)
- fnAutomatic, [507](#)
- fnFixed, [507](#)
- fnScientific, [507](#)
- GEV, [512](#)
- Guru, [512](#)
- HexNumber, [511](#)
- idFrom, [508](#)
- idNone, [508](#)
- idTo, [508](#)
- IIDC, [512](#)
- Increasing, [512](#)
- IntegerNode, [510](#)
- intflBase, [509](#)
- intflBoolean, [509](#)
- intflCategory, [509](#)
- intflCommand, [509](#)
- intflEnumEntry, [509](#)
- intflEnumeration, [509](#)
- intflFloat, [509](#)
- intflInteger, [509](#)
- intflPort, [509](#)
- intflRegister, [509](#)
- intflString, [509](#)
- intflValue, [509](#)
- Invisible, [512](#)
- IPv4Address, [511](#)
- Linear, [511](#)
- listIncrement, [508](#)
- LittleEndian, [508](#)
- Logarithmic, [511](#)
- MACAddress, [511](#)
- NA, [507](#)
- NI, [507](#)
- No, [513](#)
- NoCache, [507](#)
- noIncrement, [508](#)
- None, [512](#)
- PortNode, [511](#)
- PureNumber, [511](#)
- RegisterNode, [511](#)
- RO, [507](#)
- RW, [507](#)
- Signed, [511](#)
- spinAccessMode, [506](#)
- spinCachingMode, [507](#)
- spinDisplayNotation, [507](#)
- spinEndianess, [507](#)
- spinIncMode, [508](#)
- spinInputDirection, [508](#)
- spinInterfaceType, [508](#)
- spinLinkType, [509](#)
- spinNameSpace, [510](#)
- spinNodeCallbackFunction, [506](#)
- spinNodeCallbackHandle, [506](#)
- spinNodeHandle, [506](#)
- spinNodeMapHandle, [506](#)
- spinNodeType, [510](#)
- spinRepresentation, [511](#)
- spinSign, [511](#)
- spinSlope, [511](#)
- spinStandardNameSpace, [512](#)
- spinVisibility, [512](#)
- spinXMLValidation, [513](#)
- spinYesNo, [513](#)
- Standard, [510](#)
- StringNode, [510](#)
- UnknownNode, [511](#)
- Unsigned, [511](#)
- USB, [512](#)
- ValueNode, [510](#)
- Varying, [512](#)
- WO, [507](#)
- WriteAround, [507](#)
- WriteThrough, [507](#)
- xvAll, [513](#)
- xvCycles, [513](#)
- xvDefault, [513](#)
- xvLoad, [513](#)
- xvSFNC, [513](#)
- Yes, [513](#)
- spinnakerLogLevel
 - SpinnakerDefsC.h, [459](#)
- SpinnakerPlatformC.h
 - SPINNAKER_API, [514](#)
- spinNameSpace
 - SpinnakerGenApiDefsC.h, [510](#)
- spinNodeCallbackFunction
 - SpinnakerGenApiDefsC.h, [506](#)
- spinNodeCallbackHandle
 - SpinnakerGenApiDefsC.h, [506](#)
- spinNodeDeregisterCallback
 - SpinnakerGenApiC.h, [483](#)
- spinNodeFromString
 - SpinnakerGenApiC.h, [483](#)

- spinNodeFromStringEx
 - SpinnakerGenApiC.h, [484](#)
- spinNodeGetAccessMode
 - SpinnakerGenApiC.h, [484](#)
- spinNodeGetCachingMode
 - SpinnakerGenApiC.h, [485](#)
- spinNodeGetDescription
 - SpinnakerGenApiC.h, [485](#)
- spinNodeGetDisplayName
 - SpinnakerGenApiC.h, [486](#)
- spinNodeGetImposedAccessMode
 - SpinnakerGenApiC.h, [486](#)
- spinNodeGetImposedVisibility
 - SpinnakerGenApiC.h, [487](#)
- spinNodeGetName
 - SpinnakerGenApiC.h, [487](#)
- spinNodeGetNameSpace
 - SpinnakerGenApiC.h, [488](#)
- spinNodeGetPollingTime
 - SpinnakerGenApiC.h, [488](#)
- spinNodeGetToolTip
 - SpinnakerGenApiC.h, [489](#)
- spinNodeGetType
 - SpinnakerGenApiC.h, [489](#)
- spinNodeGetVisibility
 - SpinnakerGenApiC.h, [490](#)
- spinNodeHandle
 - SpinnakerGenApiDefsC.h, [506](#)
- spinNodeInvalidateNode
 - SpinnakerGenApiC.h, [490](#)
- spinNodeIsAvailable
 - SpinnakerGenApiC.h, [491](#)
- spinNodeIsEqual
 - SpinnakerGenApiC.h, [491](#)
- spinNodeIsImplemented
 - SpinnakerGenApiC.h, [491](#)
- spinNodeIsReadable
 - SpinnakerGenApiC.h, [492](#)
- spinNodeIsWritable
 - SpinnakerGenApiC.h, [492](#)
- spinNodeMapGetNode
 - SpinnakerGenApiC.h, [493](#)
- spinNodeMapGetNodeByIndex
 - SpinnakerGenApiC.h, [493](#)
- spinNodeMapGetNumNodes
 - SpinnakerGenApiC.h, [494](#)
- spinNodeMapHandle
 - SpinnakerGenApiDefsC.h, [506](#)
- spinNodeMapPoll
 - SpinnakerGenApiC.h, [494](#)
- spinNodeMapReleaseNode
 - SpinnakerGenApiC.h, [495](#)
- spinNodeRegisterCallback
 - SpinnakerGenApiC.h, [495](#)
- spinNodeToString
 - SpinnakerGenApiC.h, [496](#)
- spinNodeToStringEx
 - SpinnakerGenApiC.h, [496](#)
- spinNodeType
 - SpinnakerGenApiDefsC.h, [510](#)
- spinPGMOption, [298](#)
 - binaryFile, [299](#)
 - reserved, [299](#)
- spinPixelFormatFilterEnums
 - Camera Enumerations, [106](#)
- spinPixelFormatEnums
 - Camera Enumerations, [107](#)
- spinPixelFormatInfoSelectorEnums
 - Camera Enumerations, [112](#)
- spinPixelFormatSizeEnums
 - Camera Enumerations, [118](#)
- spinPNGOption, [299](#)
 - compressionLevel, [299](#)
 - interlaced, [300](#)
 - reserved, [300](#)
- spinPPMOption, [300](#)
 - binaryFile, [300](#)
 - reserved, [301](#)
- spinRegionDestinationEnums
 - Camera Enumerations, [119](#)
- spinRegionModeEnums
 - Camera Enumerations, [119](#)
- spinRegionSelectorEnums
 - Camera Enumerations, [119](#)
- spinRegisterGet
 - SpinnakerGenApiC.h, [497](#)
- spinRegisterGetAddress
 - SpinnakerGenApiC.h, [497](#)
- spinRegisterGetEx
 - SpinnakerGenApiC.h, [498](#)
- spinRegisterGetLength
 - SpinnakerGenApiC.h, [498](#)
- spinRegisterSet
 - SpinnakerGenApiC.h, [499](#)
- spinRegisterSetEx
 - SpinnakerGenApiC.h, [499](#)
- spinRegisterSetReference
 - SpinnakerGenApiC.h, [500](#)
- spinRemovalEventFunction
 - SpinnakerDefsC.h, [455](#)
- spinRepresentation
 - SpinnakerGenApiDefsC.h, [511](#)
- spinRgbTransformLightSourceEnums
 - Camera Enumerations, [120](#)
- spinScan3dCoordinateReferenceSelectorEnums
 - Camera Enumerations, [120](#)
- spinScan3dCoordinateSelectorEnums
 - Camera Enumerations, [121](#)
- spinScan3dCoordinateSystemEnums
 - Camera Enumerations, [121](#)
- spinScan3dCoordinateSystemReferenceEnums
 - Camera Enumerations, [121](#)
- spinScan3dCoordinateTransformSelectorEnums
 - Camera Enumerations, [122](#)
- spinScan3dDistanceUnitEnums
 - Camera Enumerations, [122](#)

- spinScan3dOutputModeEnums
 - Camera Enumerations, [122](#)
- spinSensorDigitizationTapsEnums
 - Camera Enumerations, [124](#)
- spinSensorShutterModeEnums
 - Camera Enumerations, [125](#)
- spinSensorTapsEnums
 - Camera Enumerations, [125](#)
- spinSequencerConfigurationModeEnums
 - Camera Enumerations, [125](#)
- spinSequencerConfigurationValidEnums
 - Camera Enumerations, [126](#)
- spinSequencerModeEnums
 - Camera Enumerations, [126](#)
- spinSequencerSetValidEnums
 - Camera Enumerations, [126](#)
- spinSequencerTriggerActivationEnums
 - Camera Enumerations, [127](#)
- spinSequencerTriggerSourceEnums
 - Camera Enumerations, [127](#)
- spinSerialPortBaudRateEnums
 - Camera Enumerations, [127](#)
- spinSerialPortParityEnums
 - Camera Enumerations, [128](#)
- spinSerialPortSelectorEnums
 - Camera Enumerations, [128](#)
- spinSerialPortSourceEnums
 - Camera Enumerations, [129](#)
- spinSerialPortStopBitsEnums
 - Camera Enumerations, [129](#)
- spinSign
 - SpinnakerGenApiDefsC.h, [511](#)
- spinSlope
 - SpinnakerGenApiDefsC.h, [511](#)
- spinSoftwareSignalSelectorEnums
 - Camera Enumerations, [129](#)
- spinSourceSelectorEnums
 - Camera Enumerations, [130](#)
- spinStandardNameSpace
 - SpinnakerGenApiDefsC.h, [512](#)
- spinStatisticsChannel
 - SpinnakerDefsC.h, [460](#)
- spinStringGetMaxLength
 - SpinnakerGenApiC.h, [500](#)
- spinStringGetValue
 - SpinnakerGenApiC.h, [501](#)
- spinStringGetValueEx
 - SpinnakerGenApiC.h, [501](#)
- spinStringSetValue
 - SpinnakerGenApiC.h, [502](#)
- spinStringSetValueEx
 - SpinnakerGenApiC.h, [502](#)
- spinSystem
 - SpinnakerDefsC.h, [455](#)
- spinSystemGetCameras
 - SpinnakerC.h, [435](#)
- spinSystemGetCamerasEx
 - SpinnakerC.h, [436](#)
- spinSystemGetInstance
 - SpinnakerC.h, [436](#)
- spinSystemGetInterfaces
 - SpinnakerC.h, [437](#)
- spinSystemGetLibraryVersion
 - SpinnakerC.h, [437](#)
- spinSystemGetLoggingLevel
 - SpinnakerC.h, [437](#)
- spinSystemGetTLNodeMap
 - SpinnakerC.h, [438](#)
- spinSystemIsInUse
 - SpinnakerC.h, [438](#)
- spinSystemRegisterDeviceArrivalEventHandler
 - SpinnakerC.h, [439](#)
- spinSystemRegisterDeviceRemovalEventHandler
 - SpinnakerC.h, [439](#)
- spinSystemRegisterInterfaceEventHandler
 - SpinnakerC.h, [440](#)
- spinSystemRegisterLogEventHandler
 - SpinnakerC.h, [440](#)
- spinSystemReleaseInstance
 - SpinnakerC.h, [441](#)
- spinSystemSendActionCommand
 - SpinnakerC.h, [441](#)
- spinSystemSetLoggingLevel
 - SpinnakerC.h, [442](#)
- spinSystemUnregisterAllLogEventHandlers
 - SpinnakerC.h, [443](#)
- spinSystemUnregisterDeviceArrivalEventHandler
 - SpinnakerC.h, [443](#)
- spinSystemUnregisterDeviceRemovalEventHandler
 - SpinnakerC.h, [444](#)
- spinSystemUnregisterInterfaceEventHandler
 - SpinnakerC.h, [444](#)
- spinSystemUnregisterLogEventHandler
 - SpinnakerC.h, [445](#)
- spinSystemUpdateCameras
 - SpinnakerC.h, [445](#)
- spinSystemUpdateCamerasEx
 - SpinnakerC.h, [446](#)
- spinTestPatternEnums
 - Camera Enumerations, [130](#)
- spinTestPatternGeneratorSelectorEnums
 - Camera Enumerations, [130](#)
- spinTIFFCompressionMethod
 - SpinnakerDefsC.h, [460](#)
- spinTIFFOption, [301](#)
 - compression, [301](#)
 - reserved, [301](#)
- spinTimerSelectorEnums
 - Camera Enumerations, [131](#)
- spinTimerStatusEnums
 - Camera Enumerations, [131](#)
- spinTimerTriggerActivationEnums
 - Camera Enumerations, [131](#)
- spinTimerTriggerSourceEnums
 - Camera Enumerations, [132](#)
- spinTLDeviceAccessStatusEnums

- Transport Layer Enumerations, [153](#)
- spinTLDeviceCurrentSpeedEnums
 - Transport Layer Enumerations, [153](#)
- spinTLDeviceEndiannessMechanismEnums
 - Transport Layer Enumerations, [154](#)
- spinTLDeviceTypeEnums
 - Transport Layer Enumerations, [154](#)
- spinTLFLIRFilterDriverStatusEnums
 - Transport Layer Enumerations, [154](#)
- spinTLGenICamXMLLocationEnums
 - Transport Layer Enumerations, [155](#)
- spinTLGevCCPEnums
 - Transport Layer Enumerations, [155](#)
- spinTLGUIXMLLocationEnums
 - Transport Layer Enumerations, [155](#)
- spinTLInterfaceTypeEnums
 - Transport Layer Enumerations, [156](#)
- spinTLPayloadType
 - SpinnakerDefsC.h, [461](#)
- spinTLPixelFormatNamespace
 - SpinnakerDefsC.h, [461](#)
- spinTLPOEStatusEnums
 - Transport Layer Enumerations, [156](#)
- spinTLStreamBufferCountModeEnums
 - Transport Layer Enumerations, [156](#)
- spinTLStreamBufferHandlingModeEnums
 - Transport Layer Enumerations, [157](#)
- spinTLStreamModeEnums
 - Transport Layer Enumerations, [157](#)
- spinTLStreamTypeEnums
 - Transport Layer Enumerations, [158](#)
- spinTLTeledyneGigeVisionFilterDriverStatusEnums
 - Transport Layer Enumerations, [158](#)
- spinTLTLTypeEnums
 - Transport Layer Enumerations, [158](#)
- spinTransferComponentSelectorEnums
 - Camera Enumerations, [133](#)
- spinTransferControlModeEnums
 - Camera Enumerations, [133](#)
- spinTransferOperationModeEnums
 - Camera Enumerations, [134](#)
- spinTransferQueueModeEnums
 - Camera Enumerations, [134](#)
- spinTransferSelectorEnums
 - Camera Enumerations, [134](#)
- spinTransferStatusSelectorEnums
 - Camera Enumerations, [135](#)
- spinTransferTriggerActivationEnums
 - Camera Enumerations, [135](#)
- spinTransferTriggerModeEnums
 - Camera Enumerations, [135](#)
- spinTransferTriggerSelectorEnums
 - Camera Enumerations, [136](#)
- spinTransferTriggerSourceEnums
 - Camera Enumerations, [136](#)
- spinTriggerActivationEnums
 - Camera Enumerations, [137](#)
- spinTriggerModeEnums
 - Camera Enumerations, [138](#)
- spinTriggerOverlapEnums
 - Camera Enumerations, [138](#)
- spinTriggerSelectorEnums
 - Camera Enumerations, [138](#)
- spinTriggerSourceEnums
 - Camera Enumerations, [138](#)
- spinUserOutputSelectorEnums
 - Camera Enumerations, [139](#)
- spinUserSetDefaultEnums
 - Camera Enumerations, [139](#)
- spinUserSetSelectorEnums
 - Camera Enumerations, [140](#)
- spinVideo
 - SpinnakerDefsC.h, [456](#)
- SpinVideo Recording Access, [151](#)
- spinVideoAppend
 - SpinVideoC.h, [515](#)
- SpinVideoC.h
 - spinVideoAppend, [515](#)
 - spinVideoClose, [515](#)
 - spinVideoOpenH264, [516](#)
 - spinVideoOpenMJPEG, [516](#)
 - spinVideoOpenUncompressed, [516](#)
 - spinVideoSetMaximumFileSize, [516](#)
- spinVideoClose
 - SpinVideoC.h, [515](#)
- spinVideoOpenH264
 - SpinVideoC.h, [516](#)
- spinVideoOpenMJPEG
 - SpinVideoC.h, [516](#)
- spinVideoOpenUncompressed
 - SpinVideoC.h, [516](#)
- spinVideoSetMaximumFileSize
 - SpinVideoC.h, [516](#)
- spinVisibility
 - SpinnakerGenApiDefsC.h, [512](#)
- spinWhiteClipSelectorEnums
 - Camera Enumerations, [140](#)
- spinXMLValidation
 - SpinnakerGenApiDefsC.h, [513](#)
- spinYesNo
 - SpinnakerGenApiDefsC.h, [513](#)
- Standard
 - SpinnakerGenApiDefsC.h, [510](#)
- Status
 - actionCommandResult, [161](#)
- StreamAnnounceBufferMinimum
 - quickSpinTLStream, [274](#)
- StreamAnnouncedBufferCount
 - quickSpinTLStream, [274](#)
- StreamBlocksProcessingTimeLast
 - quickSpinTLStream, [274](#)
- StreamBlocksProcessingTimeMax
 - quickSpinTLStream, [274](#)
- StreamBlocksProcessingTimeMin
 - quickSpinTLStream, [275](#)
- StreamBlocksReceptionTimeLast

- quickSpinTLStream, [275](#)
- StreamBlocksReceptionTimeMax
 - quickSpinTLStream, [275](#)
- StreamBlocksReceptionTimeMin
 - quickSpinTLStream, [275](#)
- StreamBlockTransferSize
 - quickSpinTLStream, [275](#)
- StreamBufferAlignment
 - quickSpinTLStream, [275](#)
- StreamBufferCountManual
 - quickSpinTLStream, [275](#)
- StreamBufferCountMax
 - quickSpinTLStream, [275](#)
- StreamBufferCountMode
 - quickSpinTLStream, [276](#)
- StreamBufferCountMode_Manual
 - Transport Layer Enumerations, [157](#)
- StreamBufferCountResult
 - quickSpinTLStream, [276](#)
- StreamBufferHandlingMode
 - quickSpinTLStream, [276](#)
- StreamBufferHandlingMode_NewestFirst
 - Transport Layer Enumerations, [157](#)
- StreamBufferHandlingMode_NewestOnly
 - Transport Layer Enumerations, [157](#)
- StreamBufferHandlingMode_OldestFirst
 - Transport Layer Enumerations, [157](#)
- StreamBufferHandlingMode_OldestFirstOverwrite
 - Transport Layer Enumerations, [157](#)
- StreamChunkCountMaximum
 - quickSpinTLStream, [276](#)
- StreamCRCCheckEnable
 - quickSpinTLStream, [276](#)
- StreamDeliveredFrameCount
 - quickSpinTLStream, [276](#)
- StreamDroppedFrameCount
 - quickSpinTLStream, [276](#)
- StreamID
 - quickSpinTLStream, [276](#)
- StreamIncompleteFrameCount
 - quickSpinTLStream, [277](#)
- StreamInputBufferCount
 - quickSpinTLStream, [277](#)
- StreamIsGrabbing
 - quickSpinTLStream, [277](#)
- StreamLostFrameCount
 - quickSpinTLStream, [277](#)
- StreamMissedPacketCount
 - quickSpinTLStream, [277](#)
- StreamMode
 - quickSpinTLStream, [277](#)
- StreamMode_LWF
 - Transport Layer Enumerations, [158](#)
- StreamMode_Socket
 - Transport Layer Enumerations, [158](#)
- StreamMode_TeledyneGigeVision
 - Transport Layer Enumerations, [158](#)
- StreamOutputBufferCount
 - quickSpinTLStream, [277](#)
- StreamPacketResendEnable
 - quickSpinTLStream, [277](#)
- StreamPacketResendMaxRequests
 - quickSpinTLStream, [278](#)
- StreamPacketResendReceivedPacketCount
 - quickSpinTLStream, [278](#)
- StreamPacketResendRequestCount
 - quickSpinTLStream, [278](#)
- StreamPacketResendRequestedPacketCount
 - quickSpinTLStream, [278](#)
- StreamPacketResendRequestTimeoutCount
 - quickSpinTLStream, [278](#)
- StreamPacketResendTimeout
 - quickSpinTLStream, [278](#)
- StreamPacketsDuplicatedCount
 - quickSpinTLStream, [278](#)
- StreamPacketsNotYetAvailableCount
 - quickSpinTLStream, [278](#)
- StreamPacketsPerFrameCount
 - quickSpinTLStream, [279](#)
- StreamPacketsTemporarilyUnavailableCount
 - quickSpinTLStream, [279](#)
- StreamPacketsTimeoutCount
 - quickSpinTLStream, [279](#)
- StreamPacketsUnavailableCount
 - quickSpinTLStream, [279](#)
- StreamReceivedFrameCount
 - quickSpinTLStream, [279](#)
- StreamReceivedPacketCount
 - quickSpinTLStream, [279](#)
- StreamStartedFrameCount
 - quickSpinTLStream, [279](#)
- StreamType
 - quickSpinTLStream, [279](#)
- StreamType_CameraLink
 - Transport Layer Enumerations, [158](#)
- StreamType_CameraLinkHS
 - Transport Layer Enumerations, [158](#)
- StreamType_CoaxPress
 - Transport Layer Enumerations, [158](#)
- StreamType_Custom
 - Transport Layer Enumerations, [158](#)
- StreamType_GigEVision
 - Transport Layer Enumerations, [158](#)
- StreamType_USB3Vision
 - Transport Layer Enumerations, [158](#)
- String Access, [149](#)
- StringNode
 - SpinnakerGenApiDefsC.h, [510](#)
- System Access, [143](#)
- TeledyneGigeVisionFilterDriverStatus
 - quickSpinTLInterface, [273](#)
- TeledyneGigeVisionFilterDriverStatus_Disabled
 - Transport Layer Enumerations, [158](#)
- TeledyneGigeVisionFilterDriverStatus_Enabled
 - Transport Layer Enumerations, [158](#)
- TeledyneGigeVisionFilterDriverStatus_NotSupported

- Transport Layer Enumerations, [158](#)
- Test0001
 - quickSpin, [250](#)
- TestEventGenerate
 - quickSpin, [250](#)
- TestPattern
 - quickSpin, [250](#)
- TestPattern_Increment
 - Camera Enumerations, [130](#)
- TestPattern_Off
 - Camera Enumerations, [130](#)
- TestPattern_SensorTestPattern
 - Camera Enumerations, [130](#)
- TestPatternGeneratorSelector
 - quickSpin, [251](#)
- TestPatternGeneratorSelector_PipelineStart
 - Camera Enumerations, [130](#)
- TestPatternGeneratorSelector_Sensor
 - Camera Enumerations, [130](#)
- TestPendingAck
 - quickSpin, [251](#)
- TimerDelay
 - quickSpin, [251](#)
- TimerDuration
 - quickSpin, [251](#)
- TimerReset
 - quickSpin, [251](#)
- TimerSelector
 - quickSpin, [251](#)
- TimerSelector_Timer0
 - Camera Enumerations, [131](#)
- TimerSelector_Timer1
 - Camera Enumerations, [131](#)
- TimerSelector_Timer2
 - Camera Enumerations, [131](#)
- TimerStatus
 - quickSpin, [251](#)
- TimerStatus_TimerActive
 - Camera Enumerations, [131](#)
- TimerStatus_TimerCompleted
 - Camera Enumerations, [131](#)
- TimerStatus_TimerIdle
 - Camera Enumerations, [131](#)
- TimerStatus_TimerTriggerWait
 - Camera Enumerations, [131](#)
- TimerTriggerActivation
 - quickSpin, [251](#)
- TimerTriggerActivation_AnyEdge
 - Camera Enumerations, [131](#)
- TimerTriggerActivation_FallingEdge
 - Camera Enumerations, [131](#)
- TimerTriggerActivation_LevelHigh
 - Camera Enumerations, [131](#)
- TimerTriggerActivation_LevelLow
 - Camera Enumerations, [131](#)
- TimerTriggerActivation_RisingEdge
 - Camera Enumerations, [131](#)
- TimerTriggerSource
 - quickSpin, [252](#)
- TimerTriggerSource_AcquisitionEnd
 - Camera Enumerations, [132](#)
- TimerTriggerSource_AcquisitionStart
 - Camera Enumerations, [132](#)
- TimerTriggerSource_AcquisitionTrigger
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Action0
 - Camera Enumerations, [133](#)
- TimerTriggerSource_Action1
 - Camera Enumerations, [133](#)
- TimerTriggerSource_Action2
 - Camera Enumerations, [133](#)
- TimerTriggerSource_Counter0End
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Counter0Start
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Counter1End
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Counter1Start
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Counter2End
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Counter2Start
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Encoder0
 - Camera Enumerations, [133](#)
- TimerTriggerSource_Encoder1
 - Camera Enumerations, [133](#)
- TimerTriggerSource_Encoder2
 - Camera Enumerations, [133](#)
- TimerTriggerSource_ExposureEnd
 - Camera Enumerations, [132](#)
- TimerTriggerSource_ExposureStart
 - Camera Enumerations, [132](#)
- TimerTriggerSource_FrameBurstEnd
 - Camera Enumerations, [132](#)
- TimerTriggerSource_FrameBurstStart
 - Camera Enumerations, [132](#)
- TimerTriggerSource_FrameEnd
 - Camera Enumerations, [132](#)
- TimerTriggerSource_FrameStart
 - Camera Enumerations, [132](#)
- TimerTriggerSource_FrameTrigger
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Line0
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Line1
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Line2
 - Camera Enumerations, [132](#)
- TimerTriggerSource_LineEnd
 - Camera Enumerations, [132](#)
- TimerTriggerSource_LineStart
 - Camera Enumerations, [132](#)
- TimerTriggerSource_LineTrigger
 - Camera Enumerations, [132](#)
- TimerTriggerSource_LinkTrigger0

- Camera Enumerations, [133](#)
- TimerTriggerSource_LinkTrigger1
 - Camera Enumerations, [133](#)
- TimerTriggerSource_LinkTrigger2
 - Camera Enumerations, [133](#)
- TimerTriggerSource_Off
 - Camera Enumerations, [132](#)
- TimerTriggerSource_SoftwareSignal0
 - Camera Enumerations, [133](#)
- TimerTriggerSource_SoftwareSignal1
 - Camera Enumerations, [133](#)
- TimerTriggerSource_SoftwareSignal2
 - Camera Enumerations, [133](#)
- TimerTriggerSource_Timer0End
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Timer0Start
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Timer1End
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Timer1Start
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Timer2End
 - Camera Enumerations, [132](#)
- TimerTriggerSource_Timer2Start
 - Camera Enumerations, [132](#)
- TimerTriggerSource_UserOutput0
 - Camera Enumerations, [132](#)
- TimerTriggerSource_UserOutput1
 - Camera Enumerations, [132](#)
- TimerTriggerSource_UserOutput2
 - Camera Enumerations, [132](#)
- TimerValue
 - [quickSpin](#), [252](#)
- Timestamp
 - [quickSpin](#), [252](#)
- TimestampLatch
 - [quickSpin](#), [252](#)
- TimestampLatchValue
 - [quickSpin](#), [252](#)
- TimestampReset
 - [quickSpin](#), [252](#)
- TLDevice Structures, [159](#)
- TLDisplayName
 - [quickSpinTLSystem](#), [283](#)
- TLFileName
 - [quickSpinTLSystem](#), [283](#)
- TLID
 - [quickSpinTLSystem](#), [283](#)
- TLInterface Structures, [159](#)
- TLModelName
 - [quickSpinTLSystem](#), [283](#)
- TLParamsLocked
 - [quickSpin](#), [252](#)
- TLPath
 - [quickSpinTLSystem](#), [283](#)
- TLStream Structures, [160](#)
- TLSystem Structures, [160](#)
- TLType
 - [quickSpinTLSystem](#), [283](#)
- TLType_CameraLink
 - Transport Layer Enumerations, [159](#)
- TLType_CameraLinkHS
 - Transport Layer Enumerations, [159](#)
- TLType_CoaXPRESS
 - Transport Layer Enumerations, [159](#)
- TLType_Custom
 - Transport Layer Enumerations, [159](#)
- TLType_GigEVision
 - Transport Layer Enumerations, [159](#)
- TLType_Mixed
 - Transport Layer Enumerations, [159](#)
- TLType_USB3Vision
 - Transport Layer Enumerations, [159](#)
- TLVendorName
 - [quickSpinTLSystem](#), [283](#)
- TLVersion
 - [quickSpinTLSystem](#), [284](#)
- TransferAbort
 - [quickSpin](#), [252](#)
- TransferBlockCount
 - [quickSpin](#), [253](#)
- TransferBurstCount
 - [quickSpin](#), [253](#)
- TransferComponentSelector
 - [quickSpin](#), [253](#)
- TransferComponentSelector_All
 - Camera Enumerations, [133](#)
- TransferComponentSelector_Blue
 - Camera Enumerations, [133](#)
- TransferComponentSelector_Green
 - Camera Enumerations, [133](#)
- TransferComponentSelector_Red
 - Camera Enumerations, [133](#)
- TransferControlMode
 - [quickSpin](#), [253](#)
- TransferControlMode_Automatic
 - Camera Enumerations, [134](#)
- TransferControlMode_Basic
 - Camera Enumerations, [134](#)
- TransferControlMode_UserControlled
 - Camera Enumerations, [134](#)
- TransferOperationMode
 - [quickSpin](#), [253](#)
- TransferOperationMode_Continuous
 - Camera Enumerations, [134](#)
- TransferOperationMode_MultiBlock
 - Camera Enumerations, [134](#)
- TransferPause
 - [quickSpin](#), [253](#)
- TransferQueueCurrentBlockCount
 - [quickSpin](#), [253](#)
- TransferQueueMaxBlockCount
 - [quickSpin](#), [253](#)
- TransferQueueMode
 - [quickSpin](#), [254](#)
- TransferQueueMode_FirstInFirstOut

- Camera Enumerations, [134](#)
- TransferQueueOverflowCount
 - quickSpin, [254](#)
- TransferResume
 - quickSpin, [254](#)
- TransferSelector
 - quickSpin, [254](#)
- TransferSelector_All
 - Camera Enumerations, [134](#)
- TransferSelector_Stream0
 - Camera Enumerations, [134](#)
- TransferSelector_Stream1
 - Camera Enumerations, [134](#)
- TransferSelector_Stream2
 - Camera Enumerations, [134](#)
- TransferStart
 - quickSpin, [254](#)
- TransferStatus
 - quickSpin, [254](#)
- TransferStatusSelector
 - quickSpin, [254](#)
- TransferStatusSelector_Paused
 - Camera Enumerations, [135](#)
- TransferStatusSelector_QueueOverflow
 - Camera Enumerations, [135](#)
- TransferStatusSelector_Stopped
 - Camera Enumerations, [135](#)
- TransferStatusSelector_Stopping
 - Camera Enumerations, [135](#)
- TransferStatusSelector_Streaming
 - Camera Enumerations, [135](#)
- TransferStop
 - quickSpin, [254](#)
- TransferStreamChannel
 - quickSpin, [255](#)
- TransferTriggerActivation
 - quickSpin, [255](#)
- TransferTriggerActivation_AnyEdge
 - Camera Enumerations, [135](#)
- TransferTriggerActivation_FallingEdge
 - Camera Enumerations, [135](#)
- TransferTriggerActivation_LevelHigh
 - Camera Enumerations, [135](#)
- TransferTriggerActivation_LevelLow
 - Camera Enumerations, [135](#)
- TransferTriggerActivation_RisingEdge
 - Camera Enumerations, [135](#)
- TransferTriggerMode
 - quickSpin, [255](#)
- TransferTriggerMode_Off
 - Camera Enumerations, [136](#)
- TransferTriggerMode_On
 - Camera Enumerations, [136](#)
- TransferTriggerSelector
 - quickSpin, [255](#)
- TransferTriggerSelector_TransferAbort
 - Camera Enumerations, [136](#)
- TransferTriggerSelector_TransferActive

- Camera Enumerations, [136](#)
- TransferTriggerSelector_TransferBurstStart
 - Camera Enumerations, [136](#)
- TransferTriggerSelector_TransferBurstStop
 - Camera Enumerations, [136](#)
- TransferTriggerSelector_TransferPause
 - Camera Enumerations, [136](#)
- TransferTriggerSelector_TransferResume
 - Camera Enumerations, [136](#)
- TransferTriggerSelector_TransferStart
 - Camera Enumerations, [136](#)
- TransferTriggerSelector_TransferStop
 - Camera Enumerations, [136](#)
- TransferTriggerSource
 - quickSpin, [255](#)
- TransferTriggerSource_Action0
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Action1
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Action2
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Counter0End
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Counter0Start
 - Camera Enumerations, [136](#)
- TransferTriggerSource_Counter1End
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Counter1Start
 - Camera Enumerations, [136](#)
- TransferTriggerSource_Counter2End
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Counter2Start
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Line0
 - Camera Enumerations, [136](#)
- TransferTriggerSource_Line1
 - Camera Enumerations, [136](#)
- TransferTriggerSource_Line2
 - Camera Enumerations, [136](#)
- TransferTriggerSource_SoftwareSignal0
 - Camera Enumerations, [137](#)
- TransferTriggerSource_SoftwareSignal1
 - Camera Enumerations, [137](#)
- TransferTriggerSource_SoftwareSignal2
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Timer0End
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Timer0Start
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Timer1End
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Timer1Start
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Timer2End
 - Camera Enumerations, [137](#)
- TransferTriggerSource_Timer2Start
 - Camera Enumerations, [137](#)
- Transport Layer Enumerations, [151](#)

- DeviceAccessStatus_Busy, 153
- DeviceAccessStatus_NoAccess, 153
- DeviceAccessStatus_OpenReadOnly, 153
- DeviceAccessStatus_OpenReadWrite, 153
- DeviceAccessStatus_ReadOnly, 153
- DeviceAccessStatus_ReadWrite, 153
- DeviceAccessStatus_Unknown, 153
- DeviceCurrentSpeed_FullSpeed, 154
- DeviceCurrentSpeed_HighSpeed, 154
- DeviceCurrentSpeed_LowSpeed, 154
- DeviceCurrentSpeed_SuperSpeed, 154
- DeviceCurrentSpeed_UnknownSpeed, 154
- DeviceEndiannessMechanism_Legacy, 154
- DeviceEndiannessMechanism_Standard, 154
- DeviceType_CameraLink, 154
- DeviceType_CameraLinkHS, 154
- DeviceType_CoaXPress, 154
- DeviceType_Custom, 154
- DeviceType_GigEVision, 154
- DeviceType_USB3Vision, 154
- FLIRFilterDriverStatus_Disabled, 155
- FLIRFilterDriverStatus_Enabled, 155
- FLIRFilterDriverStatus_NotSupported, 155
- GenICamXMLLocation_Device, 155
- GenICamXMLLocation_Host, 155
- GevCCP_EnumEntry_GevCCP_ControlAccess, 155
- GevCCP_EnumEntry_GevCCP_ExclusiveAccess, 155
- GevCCP_EnumEntry_GevCCP_OpenAccess, 155
- GUIXMLLocation_Device, 156
- GUIXMLLocation_Host, 156
- InterfaceType_CameraLink, 156
- InterfaceType_CameraLinkHS, 156
- InterfaceType_CoaXPress, 156
- InterfaceType_Custom, 156
- InterfaceType_GigEVision, 156
- InterfaceType_USB3Vision, 156
- NUMDEVICEACCESSSTATUS, 153
- NUMDEVICECURRENTSPEED, 154
- NUMDEVICEENDIANESSMECHANISM, 154
- NUMDEVICETYPE, 154
- NUMFLIRFILTERDRIVERSTATUS, 155
- NUMGENICAMXMLLOCATION, 155
- NUMGEVCCP, 155
- NUMGUIXMLLOCATION, 156
- NUMINTERFACETYPE, 156
- NUMPOESTATUS, 156
- NUMSTREAMBUFFERCOUNTMODE, 157
- NUMSTREAMBUFFERHANDLINGMODE, 157
- NUMSTREAMMODE, 158
- NUMSTREAMTYPE, 158
- NUMTELEDYNEGIGEVISIONFILTERDRIVER-STATUS, 158
- NUMTLTYPE, 159
- POEStatus_NotSupported, 156
- POEStatus_PowerOff, 156
- POEStatus_PowerOn, 156
- spinTLDeviceAccessStatusEnums, 153
- spinTLDeviceCurrentSpeedEnums, 153
- spinTLDeviceEndiannessMechanismEnums, 154
- spinTLDeviceTypeEnums, 154
- spinTLFLIRFilterDriverStatusEnums, 154
- spinTLGenICamXMLLocationEnums, 155
- spinTLGevCCPEnums, 155
- spinTLGUIXMLLocationEnums, 155
- spinTLInterfaceTypeEnums, 156
- spinTLPOEStatusEnums, 156
- spinTLStreamBufferCountModeEnums, 156
- spinTLStreamBufferHandlingModeEnums, 157
- spinTLStreamModeEnums, 157
- spinTLStreamTypeEnums, 158
- spinTLTeledyneGigeVisionFilterDriverStatusEnums, 158
- spinTLTLTypeEnums, 158
- StreamBufferCountMode_Manual, 157
- StreamBufferHandlingMode_NewestFirst, 157
- StreamBufferHandlingMode_NewestOnly, 157
- StreamBufferHandlingMode_OldestFirst, 157
- StreamBufferHandlingMode_OldestFirstOverwrite, 157
- StreamMode_LWF, 158
- StreamMode_Socket, 158
- StreamMode_TeledyneGigeVision, 158
- StreamType_CameraLink, 158
- StreamType_CameraLinkHS, 158
- StreamType_CoaXPress, 158
- StreamType_Custom, 158
- StreamType_GigEVision, 158
- StreamType_USB3Vision, 158
- TeledyneGigeVisionFilterDriverStatus_Disabled, 158
- TeledyneGigeVisionFilterDriverStatus_Enabled, 158
- TeledyneGigeVisionFilterDriverStatus_NotSupported, 158
- TLType_CameraLink, 159
- TLType_CameraLinkHS, 159
- TLType_CoaXPress, 159
- TLType_Custom, 159
- TLType_GigEVision, 159
- TLType_Mixed, 159
- TLType_USB3Vision, 159
- TriggerActivation
 - quickSpin, 255
- TriggerActivation_AnyEdge
 - Camera Enumerations, 137
- TriggerActivation_FallingEdge
 - Camera Enumerations, 137
- TriggerActivation_LevelHigh
 - Camera Enumerations, 137
- TriggerActivation_LevelLow
 - Camera Enumerations, 137
- TriggerActivation_RisingEdge
 - Camera Enumerations, 137
- TriggerDelay

- quickSpin, [255](#)
- TriggerDivider
 - quickSpin, [255](#)
- TriggerEventTest
 - quickSpin, [256](#)
- TriggerMode
 - quickSpin, [256](#)
- TriggerMode_Off
 - Camera Enumerations, [138](#)
- TriggerMode_On
 - Camera Enumerations, [138](#)
- TriggerMultiplier
 - quickSpin, [256](#)
- TriggerOverlap
 - quickSpin, [256](#)
- TriggerOverlap_Off
 - Camera Enumerations, [138](#)
- TriggerOverlap_PreviousFrame
 - Camera Enumerations, [138](#)
- TriggerOverlap_ReadOut
 - Camera Enumerations, [138](#)
- TriggerSelector
 - quickSpin, [256](#)
- TriggerSelector_AcquisitionStart
 - Camera Enumerations, [138](#)
- TriggerSelector_FrameBurstStart
 - Camera Enumerations, [138](#)
- TriggerSelector_FrameStart
 - Camera Enumerations, [138](#)
- TriggerSoftware
 - quickSpin, [256](#)
- TriggerSource
 - quickSpin, [256](#)
- TriggerSource_Action0
 - Camera Enumerations, [139](#)
- TriggerSource_Counter0End
 - Camera Enumerations, [139](#)
- TriggerSource_Counter0Start
 - Camera Enumerations, [139](#)
- TriggerSource_Counter1End
 - Camera Enumerations, [139](#)
- TriggerSource_Counter1Start
 - Camera Enumerations, [139](#)
- TriggerSource_Line0
 - Camera Enumerations, [139](#)
- TriggerSource_Line1
 - Camera Enumerations, [139](#)
- TriggerSource_Line2
 - Camera Enumerations, [139](#)
- TriggerSource_Line3
 - Camera Enumerations, [139](#)
- TriggerSource_LogicBlock0
 - Camera Enumerations, [139](#)
- TriggerSource_LogicBlock1
 - Camera Enumerations, [139](#)
- TriggerSource_Software
 - Camera Enumerations, [139](#)
- TriggerSource_UserOutput0
 - Camera Enumerations, [139](#)
- TriggerSource_UserOutput1
 - Camera Enumerations, [139](#)
- TriggerSource_UserOutput2
 - Camera Enumerations, [139](#)
- TriggerSource_UserOutput3
 - Camera Enumerations, [139](#)
- True
 - SpinnakerDefsC.h, [462](#)
- type
 - spinLibraryVersion, [296](#)
- UNKNOWN_PIXELFORMAT
 - Camera Enumerations, [112](#)
- UnknownNode
 - SpinnakerGenApiDefsC.h, [511](#)
- Unsigned
 - SpinnakerGenApiDefsC.h, [511](#)
- USB
 - SpinnakerGenApiDefsC.h, [512](#)
- UserOutputSelector
 - quickSpin, [256](#)
- UserOutputSelector_UserOutput0
 - Camera Enumerations, [139](#)
- UserOutputSelector_UserOutput1
 - Camera Enumerations, [139](#)
- UserOutputSelector_UserOutput2
 - Camera Enumerations, [139](#)
- UserOutputSelector_UserOutput3
 - Camera Enumerations, [139](#)
- UserOutputValue
 - quickSpin, [257](#)
- UserOutputValueAll
 - quickSpin, [257](#)
- UserOutputValueAllMask
 - quickSpin, [257](#)
- UserSetDefault
 - quickSpin, [257](#)
- UserSetDefault_Default
 - Camera Enumerations, [140](#)
- UserSetDefault_UserSet0
 - Camera Enumerations, [140](#)
- UserSetDefault_UserSet1
 - Camera Enumerations, [140](#)
- UserSetFeatureEnable
 - quickSpin, [257](#)
- UserSetLoad
 - quickSpin, [257](#)
- UserSetSave
 - quickSpin, [257](#)
- UserSetSelector
 - quickSpin, [257](#)
- UserSetSelector_Default
 - Camera Enumerations, [140](#)
- UserSetSelector_UserSet0
 - Camera Enumerations, [140](#)
- UserSetSelector_UserSet1
 - Camera Enumerations, [140](#)

- V3_3Enable
 - quickSpin, [258](#)
- ValueNode
 - SpinnakerGenApiDefsC.h, [510](#)
- Varying
 - SpinnakerGenApiDefsC.h, [512](#)
- WhiteClip
 - quickSpin, [258](#)
- WhiteClipSelector
 - quickSpin, [258](#)
- WhiteClipSelector_All
 - Camera Enumerations, [140](#)
- WhiteClipSelector_Blue
 - Camera Enumerations, [140](#)
- WhiteClipSelector_Green
 - Camera Enumerations, [140](#)
- WhiteClipSelector_Red
 - Camera Enumerations, [140](#)
- WhiteClipSelector_Tap1
 - Camera Enumerations, [140](#)
- WhiteClipSelector_Tap2
 - Camera Enumerations, [140](#)
- WhiteClipSelector_U
 - Camera Enumerations, [140](#)
- WhiteClipSelector_V
 - Camera Enumerations, [140](#)
- WhiteClipSelector_Y
 - Camera Enumerations, [140](#)
- Width
 - quickSpin, [258](#)
- width
 - spinAVIOption, [285](#)
 - spinH264Option, [293](#)
 - spinMJPEGOption, [298](#)
- WidthMax
 - quickSpin, [258](#)
- WO
 - SpinnakerGenApiDefsC.h, [507](#)
- WriteAround
 - SpinnakerGenApiDefsC.h, [507](#)
- WriteThrough
 - SpinnakerGenApiDefsC.h, [507](#)
- xvAll
 - SpinnakerGenApiDefsC.h, [513](#)
- xvCycles
 - SpinnakerGenApiDefsC.h, [513](#)
- xvDefault
 - SpinnakerGenApiDefsC.h, [513](#)
- xvLoad
 - SpinnakerGenApiDefsC.h, [513](#)
- xvSFNC
 - SpinnakerGenApiDefsC.h, [513](#)
- Yes
 - SpinnakerGenApiDefsC.h, [513](#)