

FlyCapture2 Managed  
2.13.3.61

Generated by Doxygen 1.7.5

Wed Apr 3 2019 19:09:01



# Contents

<b>1</b>	<b>Software Licensing Information</b>	<b>1</b>
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Namespace Index</b>	<b>5</b>
3.1	Namespace List . . . . .	5
<b>4</b>	<b>Class Index</b>	<b>7</b>
4.1	Class Hierarchy . . . . .	7
<b>5</b>	<b>Class Index</b>	<b>9</b>
5.1	Class List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>13</b>
6.1	Enumerations . . . . .	13
6.1.1	Enumeration Type Documentation . . . . .	15
6.1.1.1	BandwidthAllocation . . . . .	15
6.1.1.2	BayerTileFormat . . . . .	15
6.1.1.3	BusSpeed . . . . .	16
6.1.1.4	ByteOrder . . . . .	16
6.1.1.5	ColorProcessingAlgorithm . . . . .	16
6.1.1.6	DriverType . . . . .	17
6.1.1.7	ErrorType . . . . .	17
6.1.1.8	FrameRate . . . . .	19
6.1.1.9	GigEPropertyType . . . . .	19
6.1.1.10	GrabMode . . . . .	19

6.1.1.11	GrabTimeout	20
6.1.1.12	ImageFileFormat	20
6.1.1.13	InterfaceType	21
6.1.1.14	ManagedCallbackType	21
6.1.1.15	Mode	21
6.1.1.16	OSType	22
6.1.1.17	PCleBusSpeed	23
6.1.1.18	PixelFormat	23
6.1.1.19	PropertyType	24
6.1.1.20	StatisticsChannel	24
6.1.1.21	VideoMode	25
6.2	Structures	26
6.3	Image saving structures.	28
6.3.1	Detailed Description	28
<b>7</b>	<b>Namespace Documentation</b>	<b>29</b>
7.1	FlyCapture2 Namespace Reference	29
7.2	FlyCapture2Managed Namespace Reference	29
7.2.1	Function Documentation	34
7.2.1.1	AsyncCommandCallback	34
7.2.1.2	CommandCallbackDelegate	34
7.2.1.3	EnumCallback	34
7.2.1.4	htonl	34
7.2.1.5	ImageCallbackDelegate	34
7.2.1.6	ImageEventCallback	34
7.2.1.7	ManagedCameraEventCallback	35
7.2.1.8	ManagedCameraEventCallbackDelegate	35
7.3	FlyCapture2Managed::Gui Namespace Reference	35
<b>8</b>	<b>Class Documentation</b>	<b>37</b>
8.1	BMPOption Struct Reference	37
8.1.1	Detailed Description	37
8.1.2	Constructor & Destructor Documentation	37
8.1.2.1	BMPOption	37
8.1.3	Property Documentation	37

8.1.3.1	<a href="#">indexedColor_8bit</a>	37
8.2	<a href="#">CameraControlDialog Class Reference</a>	37
8.2.1	<a href="#">Detailed Description</a>	38
8.2.2	<a href="#">Constructor &amp; Destructor Documentation</a>	38
8.2.2.1	<a href="#">CameraControlDialog</a>	38
8.2.2.2	<a href="#">~CameraControlDialog</a>	38
8.2.3	<a href="#">Member Function Documentation</a>	38
8.2.3.1	<a href="#">Connect</a>	38
8.2.3.2	<a href="#">Disconnect</a>	38
8.2.3.3	<a href="#">Hide</a>	38
8.2.3.4	<a href="#">IsVisible</a>	38
8.2.3.5	<a href="#">SetTitle</a>	38
8.2.3.6	<a href="#">Show</a>	38
8.3	<a href="#">CameraInfo Struct Reference</a>	38
8.3.1	<a href="#">Detailed Description</a>	40
8.3.2	<a href="#">Property Documentation</a>	40
8.3.2.1	<a href="#">applicationIPAddress</a>	40
8.3.2.2	<a href="#">applicationPort</a>	40
8.3.2.3	<a href="#">bayerTileFormat</a>	40
8.3.2.4	<a href="#">busNumber</a>	40
8.3.2.5	<a href="#">ccpStatus</a>	40
8.3.2.6	<a href="#">configROM</a>	41
8.3.2.7	<a href="#">defaultGateway</a>	41
8.3.2.8	<a href="#">driverName</a>	41
8.3.2.9	<a href="#">driverType</a>	41
8.3.2.10	<a href="#">firmwareBuildTime</a>	41
8.3.2.11	<a href="#">firmwareVersion</a>	41
8.3.2.12	<a href="#">gigEMajorVersion</a>	41
8.3.2.13	<a href="#">gigEMinorVersion</a>	41
8.3.2.14	<a href="#">iidxVersion</a>	41
8.3.2.15	<a href="#">interfaceType</a>	41
8.3.2.16	<a href="#">ipAddress</a>	42
8.3.2.17	<a href="#">isColorCamera</a>	42
8.3.2.18	<a href="#">macAddress</a>	42

8.3.2.19	maximumBusSpeed	42
8.3.2.20	modelName	42
8.3.2.21	nodeNumber	42
8.3.2.22	pcieBusSpeed	42
8.3.2.23	sensorInfo	42
8.3.2.24	sensorResolution	42
8.3.2.25	serialNumber	42
8.3.2.26	subnetMask	43
8.3.2.27	userDefinedName	43
8.3.2.28	vendorName	43
8.3.2.29	xmlURL1	43
8.3.2.30	xmlURL2	43
8.4	CameraProperty Struct Reference	43
8.4.1	Detailed Description	44
8.4.2	Constructor & Destructor Documentation	44
8.4.2.1	CameraProperty	44
8.4.2.2	CameraProperty	44
8.4.3	Property Documentation	44
8.4.3.1	absControl	44
8.4.3.2	absValue	44
8.4.3.3	autoManualMode	44
8.4.3.4	onePush	44
8.4.3.5	onOff	44
8.4.3.6	present	45
8.4.3.7	type	45
8.4.3.8	valueA	45
8.4.3.9	valueB	45
8.5	CameraPropertyInfo Struct Reference	45
8.5.1	Detailed Description	46
8.5.2	Constructor & Destructor Documentation	46
8.5.2.1	CameraPropertyInfo	46
8.5.2.2	CameraPropertyInfo	46
8.5.3	Property Documentation	46
8.5.3.1	absMax	46

8.5.3.2	<a href="#">absMin</a>	46
8.5.3.3	<a href="#">absValSupported</a>	46
8.5.3.4	<a href="#">autoSupported</a>	47
8.5.3.5	<a href="#">manualSupported</a>	47
8.5.3.6	<a href="#">max</a>	47
8.5.3.7	<a href="#">min</a>	47
8.5.3.8	<a href="#">onePushSupported</a>	47
8.5.3.9	<a href="#">onOffSupported</a>	47
8.5.3.10	<a href="#">present</a>	47
8.5.3.11	<a href="#">readOutSupported</a>	47
8.5.3.12	<a href="#">type</a>	47
8.5.3.13	<a href="#">unitAbbr</a>	47
8.5.3.14	<a href="#">units</a>	48
8.6	<a href="#">CameraSelectionDialog Class Reference</a>	48
8.6.1	<a href="#">Detailed Description</a>	48
8.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	48
8.6.2.1	<a href="#">CameraSelectionDialog</a>	48
8.6.2.2	<a href="#">~CameraSelectionDialog</a>	48
8.6.3	<a href="#">Member Function Documentation</a>	48
8.6.3.1	<a href="#">GetSelectedCameraGuids</a>	48
8.6.3.2	<a href="#">SetTitle</a>	49
8.6.3.3	<a href="#">ShowModal</a>	49
8.7	<a href="#">CameraStats Struct Reference</a>	49
8.7.1	<a href="#">Detailed Description</a>	50
8.7.2	<a href="#">Constructor &amp; Destructor Documentation</a>	50
8.7.2.1	<a href="#">CameraStats</a>	50
8.7.3	<a href="#">Property Documentation</a>	50
8.7.3.1	<a href="#">cameraCurrents</a>	50
8.7.3.2	<a href="#">cameraPowerUp</a>	50
8.7.3.3	<a href="#">cameraVoltages</a>	50
8.7.3.4	<a href="#">imageCorrupt</a>	50
8.7.3.5	<a href="#">imageDriverDropped</a>	50
8.7.3.6	<a href="#">imageDropped</a>	50
8.7.3.7	<a href="#">imageXmitFailed</a>	50

8.7.3.8	numCurrents	50
8.7.3.9	numResendPacketsReceived	50
8.7.3.10	numResendPacketsRequested	50
8.7.3.11	numVoltages	50
8.7.3.12	portErrors	50
8.7.3.13	regReadFailed	50
8.7.3.14	regWriteFailed	50
8.7.3.15	temperature	51
8.7.3.16	timeSinceBusReset	51
8.7.3.17	timeSinceInitialization	51
8.7.3.18	timeStamp	51
8.8	ConfigROM Struct Reference	51
8.8.1	Detailed Description	51
8.8.2	Property Documentation	52
8.8.2.1	chipIdHi	52
8.8.2.2	chipIdLo	52
8.8.2.3	keyword	52
8.8.2.4	nodeVendorId	52
8.8.2.5	unitSpecId	52
8.8.2.6	unitSubSWVer	52
8.8.2.7	unitSWVer	52
8.8.2.8	vendorUniqueInfo0	52
8.8.2.9	vendorUniqueInfo1	52
8.8.2.10	vendorUniqueInfo2	52
8.8.2.11	vendorUniqueInfo3	53
8.9	EmbeddedImageInfo Struct Reference	53
8.9.1	Detailed Description	53
8.9.2	Constructor & Destructor Documentation	53
8.9.2.1	EmbeddedImageInfo	53
8.9.3	Property Documentation	53
8.9.3.1	brightness	53
8.9.3.2	exposure	53
8.9.3.3	frameCounter	53
8.9.3.4	gain	53



8.9.3.5	GPIOPinState	54
8.9.3.6	ROIPosition	54
8.9.3.7	shutter	54
8.9.3.8	strobePattern	54
8.9.3.9	timestamp	54
8.9.3.10	whiteBalance	54
8.10	EmbeddedImageInfoProperty Struct Reference	54
8.10.1	Detailed Description	54
8.10.2	Property Documentation	54
8.10.2.1	available	54
8.10.2.2	onOff	54
8.11	FC2Config Struct Reference	54
8.11.1	Detailed Description	55
8.11.2	Constructor & Destructor Documentation	55
8.11.2.1	FC2Config	55
8.11.3	Property Documentation	55
8.11.3.1	asyncBusSpeed	56
8.11.3.2	bandwidthAllocation	56
8.11.3.3	grabMode	56
8.11.3.4	grabTimeout	56
8.11.3.5	highPerformanceRetrieveBuffer	56
8.11.3.6	isochBusSpeed	56
8.11.3.7	minNumImageNotifications	56
8.11.3.8	numBuffers	57
8.11.3.9	numImageNotifications	57
8.11.3.10	registerTimeout	57
8.11.3.11	registerTimeoutRetries	57
8.12	FC2Exception Class Reference	57
8.12.1	Detailed Description	58
8.12.2	Constructor & Destructor Documentation	58
8.12.2.1	FC2Exception	58
8.12.2.2	FC2Exception	58
8.12.2.3	FC2Exception	58
8.12.2.4	~FC2Exception	58

8.12.2.5	FC2Exception	58
8.12.2.6	FC2Exception	58
8.12.3	Property Documentation	58
8.12.3.1	CauseType	58
8.12.3.2	NativeErrorTrace	58
8.12.3.3	Type	58
8.13	FC2Version Struct Reference	59
8.13.1	Detailed Description	59
8.13.2	Property Documentation	59
8.13.2.1	build	59
8.13.2.2	major	59
8.13.2.3	minor	59
8.13.2.4	type	59
8.14	Format7ImageSettings Struct Reference	59
8.14.1	Detailed Description	60
8.14.2	Property Documentation	60
8.14.2.1	height	60
8.14.2.2	mode	60
8.14.2.3	offsetX	60
8.14.2.4	offsetY	60
8.14.2.5	pixelFormat	60
8.14.2.6	width	61
8.15	Format7Info Struct Reference	61
8.15.1	Detailed Description	61
8.15.2	Property Documentation	62
8.15.2.1	imageHStepSize	62
8.15.2.2	imageVStepSize	62
8.15.2.3	maxHeight	62
8.15.2.4	maxPacketSize	62
8.15.2.5	maxWidth	62
8.15.2.6	minPacketSize	62
8.15.2.7	mode	62
8.15.2.8	offsetHStepSize	62
8.15.2.9	offsetVStepSize	62

8.15.2.10 packetSize . . . . .	62
8.15.2.11 percentage . . . . .	63
8.15.2.12 pixelFormatBitField . . . . .	63
8.15.2.13 vendorPixelFormatBitField . . . . .	63
8.16 Format7PacketInfo Struct Reference . . . . .	63
8.16.1 Detailed Description . . . . .	63
8.16.2 Property Documentation . . . . .	63
8.16.2.1 maxBytesPerPacket . . . . .	63
8.16.2.2 recommendedBytesPerPacket . . . . .	63
8.16.2.3 unitBytesPerPacket . . . . .	63
8.17 GigEConfig Struct Reference . . . . .	64
8.17.1 Detailed Description . . . . .	64
8.17.2 Property Documentation . . . . .	64
8.17.2.1 enablePacketResend . . . . .	64
8.18 GigEImageSettings Struct Reference . . . . .	64
8.18.1 Detailed Description . . . . .	64
8.18.2 Property Documentation . . . . .	65
8.18.2.1 height . . . . .	65
8.18.2.2 offsetX . . . . .	65
8.18.2.3 offsetY . . . . .	65
8.18.2.4 pixelFormat . . . . .	65
8.18.2.5 width . . . . .	65
8.19 GigEImageSettingsInfo Struct Reference . . . . .	65
8.19.1 Detailed Description . . . . .	66
8.19.2 Property Documentation . . . . .	66
8.19.2.1 imageHStepSize . . . . .	66
8.19.2.2 imageVStepSize . . . . .	66
8.19.2.3 maxHeight . . . . .	66
8.19.2.4 maxWidth . . . . .	66
8.19.2.5 offsetHStepSize . . . . .	66
8.19.2.6 offsetVStepSize . . . . .	66
8.19.2.7 pixelFormatBitField . . . . .	66
8.19.2.8 vendorPixelFormatBitField . . . . .	66
8.20 GigEProperty Struct Reference . . . . .	67

8.20.1 Detailed Description . . . . .	67
8.20.2 Property Documentation . . . . .	67
8.20.2.1 isReadable . . . . .	67
8.20.2.2 isWritable . . . . .	67
8.20.2.3 max . . . . .	67
8.20.2.4 min . . . . .	67
8.20.2.5 propType . . . . .	68
8.20.2.6 value . . . . .	68
8.21 GigEStreamChannel Struct Reference . . . . .	68
8.21.1 Detailed Description . . . . .	68
8.21.2 Property Documentation . . . . .	68
8.21.2.1 destinationIpAddress . . . . .	68
8.21.2.2 doNotFragment . . . . .	69
8.21.2.3 hostPort . . . . .	69
8.21.2.4 interPacketDelay . . . . .	69
8.21.2.5 networkInterfaceIndex . . . . .	69
8.21.2.6 packetSize . . . . .	69
8.21.2.7 sourcePort . . . . .	69
8.22 ImageMetadata Struct Reference . . . . .	69
8.22.1 Detailed Description . . . . .	70
8.22.2 Property Documentation . . . . .	70
8.22.2.1 embeddedBrightness . . . . .	70
8.22.2.2 embeddedExposure . . . . .	70
8.22.2.3 embeddedFrameCounter . . . . .	70
8.22.2.4 embeddedGain . . . . .	70
8.22.2.5 embeddedGPIOPinState . . . . .	70
8.22.2.6 embeddedROIPosition . . . . .	70
8.22.2.7 embeddedShutter . . . . .	71
8.22.2.8 embeddedStrobePattern . . . . .	71
8.22.2.9 embeddedTimeStamp . . . . .	71
8.22.2.10 embeddedWhiteBalance . . . . .	71
8.23 JpegOption Struct Reference . . . . .	71
8.23.1 Detailed Description . . . . .	71
8.23.2 Constructor & Destructor Documentation . . . . .	71

8.23.2.1	JpegOption	71
8.23.3	Property Documentation	71
8.23.3.1	progressive	72
8.23.3.2	quality	72
8.24	Jpg2Option Struct Reference	72
8.24.1	Detailed Description	72
8.24.2	Constructor & Destructor Documentation	72
8.24.2.1	Jpg2Option	72
8.24.3	Property Documentation	72
8.24.3.1	quality	72
8.25	LutData Struct Reference	73
8.25.1	Detailed Description	73
8.25.2	Property Documentation	73
8.25.2.1	enabled	73
8.25.2.2	inputBitDepth	73
8.25.2.3	numBanks	73
8.25.2.4	numChannels	73
8.25.2.5	numEntries	74
8.25.2.6	outputBitDepth	74
8.25.2.7	supported	74
8.26	ManagedBusManager Class Reference	74
8.26.1	Detailed Description	76
8.26.2	Constructor & Destructor Documentation	76
8.26.2.1	ManagedBusManager	76
8.26.2.2	~ManagedBusManager	76
8.26.2.3	!ManagedBusManager	76
8.26.3	Member Function Documentation	76
8.26.3.1	ConvertToManagedGuid	76
8.26.3.2	ConvertToNativeGuid	76
8.26.3.3	DiscoverGigECameras	77
8.26.3.4	FireBusReset	77
8.26.3.5	ForceAllIPAddressesAutomatically	77
8.26.3.6	ForceAllIPAddressesAutomatically	77
8.26.3.7	ForceIPAddressToCamera	78

8.26.3.8	GetCameraFromIndex	78
8.26.3.9	GetCameraFromIPAddress	78
8.26.3.10	GetCameraFromSerialNumber	79
8.26.3.11	GetCameraSerialNumberFromIndex	79
8.26.3.12	GetDeviceFromIndex	79
8.26.3.13	GetInterfaceTypeFromGuid	80
8.26.3.14	GetNumOfCameras	80
8.26.3.15	GetNumOfDevices	80
8.26.3.16	GetTopology	80
8.26.3.17	GetUsbLinkInfo	81
8.26.3.18	GetUsbPortStatus	81
8.26.3.19	IsCameraControlable	81
8.26.3.20	ReadPhyRegister	81
8.26.3.21	RegisterCallback	82
8.26.3.22	RescanBus	82
8.26.3.23	UnregisterCallback	82
8.26.3.24	WritePhyRegister	83
8.27	ManagedCamera Class Reference	83
8.27.1	Detailed Description	85
8.27.2	Constructor & Destructor Documentation	85
8.27.2.1	ManagedCamera	85
8.27.2.2	~ManagedCamera	85
8.27.2.3	!ManagedCamera	85
8.27.3	Member Function Documentation	86
8.27.3.1	Connect	86
8.27.3.2	GetFormat7Configuration	86
8.27.3.3	GetFormat7Info	86
8.27.3.4	GetVideoModeAndFrameRate	87
8.27.3.5	GetVideoModeAndFrameRateInfo	87
8.27.3.6	SetFormat7Configuration	88
8.27.3.7	SetFormat7Configuration	88
8.27.3.8	SetVideoModeAndFrameRate	88
8.27.3.9	StartSyncCapture	89
8.27.3.10	StartSyncCapture	89

8.27.3.11	ValidateFormat7Settings	90
8.28	ManagedCameraBase Class Reference	90
8.28.1	Detailed Description	95
8.28.2	Constructor & Destructor Documentation	95
8.28.2.1	~ManagedCameraBase	95
8.28.2.2	ManagedCameraBase	95
8.28.3	Member Function Documentation	95
8.28.3.1	Connect	95
8.28.3.2	DeregisterAllEvents	95
8.28.3.3	DeregisterEvent	95
8.28.3.4	Disconnect	95
8.28.3.5	EnableLUT	96
8.28.3.6	FireSoftwareTrigger	96
8.28.3.7	GetActiveLUTBank	96
8.28.3.8	GetCameraInfo	96
8.28.3.9	GetConfiguration	97
8.28.3.10	GetCycleTime	97
8.28.3.11	GetEmbeddedImageInfo	97
8.28.3.12	GetGPIOPinDirection	97
8.28.3.13	GetLUTBankInfo	98
8.28.3.14	GetLUTChannel	98
8.28.3.15	GetLUTInfo	98
8.28.3.16	GetMemoryChannel	99
8.28.3.17	GetMemoryChannelInfo	99
8.28.3.18	GetNativeCamera	99
8.28.3.19	GetProperty	99
8.28.3.20	GetPropertyInfo	100
8.28.3.21	GetRegisterString	100
8.28.3.22	GetStats	100
8.28.3.23	GetStrobe	101
8.28.3.24	GetStrobeInfo	101
8.28.3.25	GetTriggerDelay	101
8.28.3.26	GetTriggerDelayInfo	102
8.28.3.27	GetTriggerMode	102

8.28.3.28 GetTriggerModelInfo . . . . .	102
8.28.3.29 IsConnected . . . . .	103
8.28.3.30 OnNativeCallback . . . . .	103
8.28.3.31 OnNativeCameraEventCallback . . . . .	103
8.28.3.32 ReadRegister . . . . .	103
8.28.3.33 ReadRegisterBlock . . . . .	103
8.28.3.34 RegisterAllEvents . . . . .	104
8.28.3.35 RegisterEvent . . . . .	104
8.28.3.36 ResetStats . . . . .	104
8.28.3.37 RestoreFromMemoryChannel . . . . .	104
8.28.3.38 RetrieveBuffer . . . . .	104
8.28.3.39 SaveToMemoryChannel . . . . .	105
8.28.3.40 SetActiveLUTBank . . . . .	105
8.28.3.41 SetCallback . . . . .	105
8.28.3.42 SetCamera . . . . .	105
8.28.3.43 SetConfiguration . . . . .	106
8.28.3.44 SetEmbeddedImageInfo . . . . .	106
8.28.3.45 SetGPIOPinDirection . . . . .	106
8.28.3.46 SetGPIOPinDirection . . . . .	106
8.28.3.47 SetLUTChannel . . . . .	107
8.28.3.48 SetProperty . . . . .	107
8.28.3.49 SetProperty . . . . .	108
8.28.3.50 SetStrobe . . . . .	108
8.28.3.51 SetTriggerDelay . . . . .	108
8.28.3.52 SetTriggerDelay . . . . .	109
8.28.3.53 SetTriggerMode . . . . .	109
8.28.3.54 SetUserBuffers . . . . .	110
8.28.3.55 StartCapture . . . . .	110
8.28.3.56 StartCapture . . . . .	110
8.28.3.57 StopCapture . . . . .	111
8.28.3.58 WaitForBufferEvent . . . . .	111
8.28.3.59 WriteRegister . . . . .	111
8.28.3.60 WriteRegister . . . . .	112
8.28.3.61 WriteRegisterBlock . . . . .	112



8.28.4	Member Data Documentation . . . . .	112
8.28.4.1	m_allInternalCameraEvents . . . . .	112
8.28.4.2	m_externalDelegate . . . . .	112
8.28.4.3	m_internalCameraEventDelegate . . . . .	112
8.28.4.4	m_internalDelegate . . . . .	112
8.28.4.5	m_isLocal . . . . .	113
8.28.4.6	m_p . . . . .	113
8.28.4.7	m_pNativeCamBase . . . . .	113
8.28.4.8	m_specificInternalCameraEvents . . . . .	113
8.29	ManagedEventCallbackData Struct Reference . . . . .	113
8.29.1	Member Data Documentation . . . . .	113
8.29.1.1	EventID . . . . .	113
8.29.1.2	EventName . . . . .	113
8.29.1.3	EventTimestamp . . . . .	113
8.30	ManagedEventOptions Struct Reference . . . . .	114
8.30.1	Detailed Description . . . . .	114
8.30.2	Member Data Documentation . . . . .	114
8.30.2.1	EventCallbackFcn . . . . .	114
8.30.2.2	EventName . . . . .	114
8.31	ManagedGCCamera Class Reference . . . . .	114
8.31.1	Constructor & Destructor Documentation . . . . .	115
8.31.1.1	ManagedGCCamera . . . . .	115
8.31.1.2	~ManagedGCCamera . . . . .	115
8.31.1.3	!ManagedGCCamera . . . . .	115
8.31.2	Member Function Documentation . . . . .	115
8.31.2.1	Connect . . . . .	116
8.31.2.2	Connect . . . . .	116
8.31.2.3	Disconnect . . . . .	116
8.31.2.4	GetNodeMap . . . . .	116
8.31.2.5	SetCamera . . . . .	116
8.31.2.6	SetCamera . . . . .	116
8.32	ManagedGCPort Class Reference . . . . .	116
8.32.1	Constructor & Destructor Documentation . . . . .	117
8.32.1.1	ManagedGCPort . . . . .	117

8.32.1.2	<a href="#">~ManagedGCPort</a>	117
8.32.2	<a href="#">Member Function Documentation</a>	117
8.32.2.1	<a href="#">Read</a>	117
8.32.2.2	<a href="#">Write</a>	117
8.33	<a href="#">ManagedGigECamera Class Reference</a>	117
8.33.1	<a href="#">Detailed Description</a>	120
8.33.2	<a href="#">Constructor &amp; Destructor Documentation</a>	120
8.33.2.1	<a href="#">ManagedGigECamera</a>	120
8.33.2.2	<a href="#">~ManagedGigECamera</a>	120
8.33.2.3	<a href="#">!ManagedGigECamera</a>	120
8.33.3	<a href="#">Member Function Documentation</a>	120
8.33.3.1	<a href="#">Connect</a>	121
8.33.3.2	<a href="#">DiscoverGigEPacketSize</a>	121
8.33.3.3	<a href="#">GetGigEConfig</a>	121
8.33.3.4	<a href="#">GetGigEImageBinningSettings</a>	121
8.33.3.5	<a href="#">GetGigEImageSettings</a>	122
8.33.3.6	<a href="#">GetGigEImageSettingsInfo</a>	122
8.33.3.7	<a href="#">GetGigEImagingMode</a>	122
8.33.3.8	<a href="#">GetGigEProperty</a>	122
8.33.3.9	<a href="#">GetGigEStreamChannelInfo</a>	122
8.33.3.10	<a href="#">GetNumStreamChannels</a>	123
8.33.3.11	<a href="#">QueryGigEImagingMode</a>	123
8.33.3.12	<a href="#">ReadGVCPMemory</a>	123
8.33.3.13	<a href="#">ReadGVCPRegister</a>	123
8.33.3.14	<a href="#">ReadGVCPRegisterBlock</a>	123
8.33.3.15	<a href="#">SetGigEConfig</a>	124
8.33.3.16	<a href="#">SetGigEImageBinningSettings</a>	124
8.33.3.17	<a href="#">SetGigEImageSettings</a>	124
8.33.3.18	<a href="#">SetGigEImagingMode</a>	124
8.33.3.19	<a href="#">SetGigEProperty</a>	125
8.33.3.20	<a href="#">SetGigEStreamChannelInfo</a>	125
8.33.3.21	<a href="#">WriteGVCPMemory</a>	125
8.33.3.22	<a href="#">WriteGVCPRegister</a>	125
8.33.3.23	<a href="#">WriteGVCPRegister</a>	125

8.33.3.24	WriteGVCPRegisterBlock	126
8.34	ManagedImage Class Reference	126
8.34.1	Detailed Description	129
8.34.2	Constructor & Destructor Documentation	129
8.34.2.1	ManagedImage	129
8.34.2.2	ManagedImage	129
8.34.2.3	ManagedImage	129
8.34.2.4	ManagedImage	129
8.34.2.5	ManagedImage	129
8.34.2.6	ManagedImage	129
8.34.2.7	ManagedImage	129
8.34.2.8	ManagedImage	129
8.34.2.9	ManagedImage	129
8.34.2.10	~ManagedImage	129
8.34.2.11	ManagedImage	129
8.34.2.12	!ManagedImage	129
8.34.3	Member Function Documentation	129
8.34.3.1	CalculateStatistics	129
8.34.3.2	Convert	130
8.34.3.3	Convert	130
8.34.3.4	DetermineBitsPerPixel	130
8.34.3.5	GetDimensions	131
8.34.3.6	GetNativeImage	131
8.34.3.7	GetRawNativeImagePointer	131
8.34.3.8	IsNativeImageValid	131
8.34.3.9	ReleaseBuffer	131
8.34.3.10	Save	131
8.34.3.11	Save	131
8.34.3.12	Save	132
8.34.3.13	Save	132
8.34.3.14	Save	132
8.34.3.15	Save	132
8.34.3.16	Save	132
8.34.3.17	Save	133

8.34.3.18	Save	133
8.34.3.19	SetData	133
8.34.3.20	SetDimensions	133
8.34.4	Property Documentation	133
8.34.4.1	bayerTileFormat	134
8.34.4.2	bitmap	134
8.34.4.3	bitsPerPixel	134
8.34.4.4	blockId	134
8.34.4.5	colorProcessingAlgorithm	134
8.34.4.6	cols	134
8.34.4.7	data	134
8.34.4.8	dataSize	134
8.34.4.9	defaultColorProcessingAlgorithm	134
8.34.4.10	defaultOutputPixelFormat	135
8.34.4.11	imageMetadata	135
8.34.4.12	pixelFormat	135
8.34.4.13	receivedDataSize	135
8.34.4.14	rows	135
8.34.4.15	stride	135
8.34.4.16	timeStamp	135
8.35	ManagedImageStatistics Class Reference	136
8.35.1	Constructor & Destructor Documentation	136
8.35.1.1	ManagedImageStatistics	136
8.35.1.2	~ManagedImageStatistics	136
8.35.2	Member Function Documentation	136
8.35.2.1	DisableAll	136
8.35.2.2	EnableAll	136
8.35.2.3	EnableGreyOnly	136
8.35.2.4	EnableHSLOnly	136
8.35.2.5	EnableRGBOnly	137
8.35.2.6	GetChannelStatus	137
8.35.2.7	GetHistogram	137
8.35.2.8	GetMean	137
8.35.2.9	GetNativeImageStatistics	137

8.35.2.10	<a href="#">GetNumPixelValues</a>	137
8.35.2.11	<a href="#">GetPixelValueRange</a>	137
8.35.2.12	<a href="#">GetRange</a>	137
8.35.2.13	<a href="#">GetStatistics</a>	137
8.35.2.14	<a href="#">SetChannelStatus</a>	137
8.36	<a href="#">ManagedPGRGuid Class Reference</a>	137
8.36.1	<a href="#">Detailed Description</a>	138
8.36.2	<a href="#">Constructor &amp; Destructor Documentation</a>	138
8.36.2.1	<a href="#">ManagedPGRGuid</a>	138
8.36.2.2	<a href="#">ManagedPGRGuid</a>	138
8.36.2.3	<a href="#">ManagedPGRGuid</a>	138
8.36.3	<a href="#">Member Function Documentation</a>	138
8.36.3.1	<a href="#">Equals</a>	138
8.36.3.2	<a href="#">GetHashCode</a>	138
8.36.3.3	<a href="#">operator!=</a>	138
8.36.3.4	<a href="#">operator=</a>	139
8.36.3.5	<a href="#">operator==</a>	139
8.36.4	<a href="#">Member Data Documentation</a>	139
8.36.4.1	<a href="#">value0</a>	139
8.36.4.2	<a href="#">value1</a>	139
8.36.4.3	<a href="#">value2</a>	139
8.36.4.4	<a href="#">value3</a>	139
8.37	<a href="#">ManagedTopologyNode Class Reference</a>	139
8.37.1	<a href="#">Detailed Description</a>	140
8.37.2	<a href="#">Member Enumeration Documentation</a>	140
8.37.2.1	<a href="#">NodeType</a>	140
8.37.2.2	<a href="#">PortType</a>	141
8.37.3	<a href="#">Constructor &amp; Destructor Documentation</a>	141
8.37.3.1	<a href="#">~ManagedTopologyNode</a>	141
8.37.3.2	<a href="#">ManagedTopologyNode</a>	141
8.37.3.3	<a href="#">ManagedTopologyNode</a>	141
8.37.3.4	<a href="#">ManagedTopologyNode</a>	141
8.37.4	<a href="#">Member Function Documentation</a>	141
8.37.4.1	<a href="#">GetChild</a>	141

8.37.4.2	<a href="#">GetDeviceId</a>	141
8.37.4.3	<a href="#">GetGuid</a>	141
8.37.4.4	<a href="#">GetInterfaceType</a>	142
8.37.4.5	<a href="#">GetNodeType</a>	142
8.37.4.6	<a href="#">GetNumChildren</a>	142
8.37.4.7	<a href="#">GetNumPorts</a>	142
8.37.4.8	<a href="#">GetPortType</a>	142
8.37.4.9	<a href="#">TranslateNodeType</a>	143
8.37.4.10	<a href="#">TranslateNodeType</a>	143
8.37.4.11	<a href="#">TranslatePortType</a>	143
8.37.4.12	<a href="#">TranslatePortType</a>	143
8.38	<a href="#">ManagedUtilities Class Reference</a>	143
8.38.1	<a href="#">Member Function Documentation</a>	144
8.38.1.1	<a href="#">CheckDriver</a>	144
8.38.1.2	<a href="#">GetDriverDeviceName</a>	144
8.38.1.3	<a href="#">LaunchBrowser</a>	144
8.38.1.4	<a href="#">LaunchCommand</a>	144
8.38.1.5	<a href="#">LaunchCommandAsync</a>	144
8.38.1.6	<a href="#">LaunchHelp</a>	144
8.38.1.7	<a href="#">OnNativeCallback</a>	144
8.38.2	<a href="#">Member Data Documentation</a>	144
8.38.2.1	<a href="#">m_externalDelegate</a>	144
8.38.2.2	<a href="#">m_internalDelegate</a>	144
8.38.3	<a href="#">Property Documentation</a>	144
8.38.3.1	<a href="#">libraryVersion</a>	144
8.38.3.2	<a href="#">systemInfo</a>	144
8.39	<a href="#">NativeEventStruct Struct Reference</a>	144
8.39.1	<a href="#">Member Data Documentation</a>	144
8.39.1.1	<a href="#">ptr</a>	144
8.40	<a href="#">PgmOption Struct Reference</a>	144
8.40.1	<a href="#">Detailed Description</a>	145
8.40.2	<a href="#">Constructor &amp; Destructor Documentation</a>	145
8.40.2.1	<a href="#">PgmOption</a>	145
8.40.3	<a href="#">Property Documentation</a>	145

8.40.3.1	binaryFile	145
8.41	PngOption Struct Reference	145
8.41.1	Detailed Description	145
8.41.2	Constructor & Destructor Documentation	146
8.41.2.1	PngOption	146
8.41.3	Property Documentation	146
8.41.3.1	compressionLevel	146
8.41.3.2	interlaced	146
8.42	PpmOption Struct Reference	146
8.42.1	Detailed Description	146
8.42.2	Constructor & Destructor Documentation	146
8.42.2.1	PpmOption	146
8.42.3	Property Documentation	146
8.42.3.1	binaryFile	146
8.43	StrobeControl Struct Reference	147
8.43.1	Detailed Description	147
8.43.2	Property Documentation	147
8.43.2.1	delay	147
8.43.2.2	duration	147
8.43.2.3	onOff	147
8.43.2.4	polarity	147
8.43.2.5	source	147
8.44	StrobeInfo Struct Reference	148
8.44.1	Detailed Description	148
8.44.2	Property Documentation	148
8.44.2.1	maxValue	148
8.44.2.2	minValue	148
8.44.2.3	onOffSupported	148
8.44.2.4	polaritySupported	148
8.44.2.5	present	149
8.44.2.6	readOutSupported	149
8.44.2.7	source	149
8.45	SystemInfo Struct Reference	149
8.45.1	Detailed Description	150

8.45.2	Property Documentation	150
8.45.2.1	byteOrder	150
8.45.2.2	cpuDescription	150
8.45.2.3	driverList	150
8.45.2.4	gpuDescription	150
8.45.2.5	libraryList	150
8.45.2.6	numCpuCores	150
8.45.2.7	osDescription	150
8.45.2.8	osType	150
8.45.2.9	screenHeight	150
8.45.2.10	screenWidth	151
8.45.2.11	systemMemorySize	151
8.46	TiffOption Struct Reference	151
8.46.1	Detailed Description	151
8.46.2	Member Enumeration Documentation	151
8.46.2.1	CompressionMethod	151
8.46.3	Constructor & Destructor Documentation	152
8.46.3.1	TiffOption	152
8.46.4	Property Documentation	152
8.46.4.1	compression	152
8.47	TimeStamp Struct Reference	152
8.47.1	Detailed Description	152
8.47.2	Property Documentation	152
8.47.2.1	cycleCount	152
8.47.2.2	cycleOffset	153
8.47.2.3	cycleSeconds	153
8.47.2.4	microSeconds	153
8.47.2.5	seconds	153
8.48	Translate Class Reference	153
8.48.1	Member Function Documentation	156
8.48.1.1	ToMgd	156
8.48.1.2	ToMgd	156
8.48.1.3	ToMgd	156
8.48.1.4	ToMgd	156



8.48.1.5 ToMgd . . . . .	156
8.48.1.6 ToMgd . . . . .	156
8.48.1.7 ToMgd . . . . .	156
8.48.1.8 ToMgd . . . . .	156
8.48.1.9 ToMgd . . . . .	157
8.48.1.10 ToMgd . . . . .	157
8.48.1.11 ToMgd . . . . .	157
8.48.1.12 ToMgd . . . . .	157
8.48.1.13 ToMgd . . . . .	157
8.48.1.14 ToMgd . . . . .	157
8.48.1.15 ToMgd . . . . .	157
8.48.1.16 ToMgd . . . . .	157
8.48.1.17 ToMgd . . . . .	157
8.48.1.18 ToMgd . . . . .	157
8.48.1.19 ToMgd . . . . .	157
8.48.1.20 ToMgd . . . . .	157
8.48.1.21 ToMgd . . . . .	157
8.48.1.22 ToMgd . . . . .	157
8.48.1.23 ToMgd . . . . .	157
8.48.1.24 ToMgd . . . . .	157
8.48.1.25 ToMgd . . . . .	158
8.48.1.26 ToMgd . . . . .	158
8.48.1.27 ToMgd . . . . .	158
8.48.1.28 ToMgd . . . . .	158
8.48.1.29 ToMgd . . . . .	158
8.48.1.30 ToMgd . . . . .	158
8.48.1.31 ToMgd . . . . .	158
8.48.1.32 ToMgd . . . . .	158
8.48.1.33 ToMgd . . . . .	158
8.48.1.34 ToNative . . . . .	158
8.48.1.35 ToNative . . . . .	158
8.48.1.36 ToNative . . . . .	158
8.48.1.37 ToNative . . . . .	158
8.48.1.38 ToNative . . . . .	158

8.48.1.39 ToNative . . . . .	158
8.48.1.40 ToNative . . . . .	158
8.48.1.41 ToNative . . . . .	159
8.48.1.42 ToNative . . . . .	159
8.48.1.43 ToNative . . . . .	159
8.48.1.44 ToNative . . . . .	159
8.48.1.45 ToNative . . . . .	159
8.48.1.46 ToNative . . . . .	159
8.48.1.47 ToNative . . . . .	159
8.48.1.48 ToNative . . . . .	159
8.48.1.49 ToNative . . . . .	159
8.48.1.50 ToNative . . . . .	159
8.48.1.51 ToNative . . . . .	159
8.48.1.52 ToNative . . . . .	159
8.48.1.53 ToNative . . . . .	159
8.48.1.54 ToNative . . . . .	159
8.48.1.55 ToNative . . . . .	159
8.48.1.56 ToNative . . . . .	159
8.48.1.57 ToNative . . . . .	160
8.48.1.58 ToNative . . . . .	160
8.48.1.59 ToNative . . . . .	160
8.48.1.60 ToNative . . . . .	160
8.48.1.61 ToNative . . . . .	160
8.48.1.62 ToNative . . . . .	160
8.48.1.63 ToNative . . . . .	160
8.48.1.64 translate . . . . .	160
8.48.1.65 translate . . . . .	160
8.48.1.66 translate . . . . .	160
8.48.1.67 translate . . . . .	160
8.48.1.68 translate . . . . .	160
8.48.1.69 translate . . . . .	160
8.48.1.70 translate . . . . .	160
8.48.1.71 translate . . . . .	160
8.48.1.72 translate . . . . .	160

8.48.1.73 translate . . . . .	161
8.48.1.74 translate . . . . .	161
8.48.1.75 translate . . . . .	161
8.48.1.76 translate . . . . .	161
8.48.1.77 translate . . . . .	161
8.48.1.78 translate . . . . .	161
8.48.1.79 translate . . . . .	161
8.48.1.80 translate . . . . .	161
8.48.1.81 translate . . . . .	161
8.48.1.82 translate . . . . .	161
8.48.1.83 translate . . . . .	161
8.48.1.84 translate . . . . .	161
8.48.1.85 translate . . . . .	161
8.48.1.86 translate . . . . .	161
8.48.1.87 translate . . . . .	161
8.48.1.88 translate . . . . .	161
8.48.1.89 translate . . . . .	161
8.48.1.90 translate . . . . .	162
8.48.1.91 translate . . . . .	162
8.48.1.92 translate . . . . .	162
8.48.1.93 translate . . . . .	162
8.48.1.94 translate . . . . .	162
8.48.1.95 translate . . . . .	162
8.48.1.96 translate . . . . .	162
8.48.1.97 translate . . . . .	162
8.48.1.98 translate . . . . .	162
8.48.1.99 translate . . . . .	162
8.48.1.100translate . . . . .	162
8.48.1.101translate . . . . .	162
8.48.1.102translate . . . . .	162
8.48.1.103translate . . . . .	162
8.48.1.104Translate::ToMgd . . . . .	162
8.48.1.105Translate::ToNative . . . . .	162
8.49 TriggerMode Struct Reference . . . . .	163

8.49.1 Detailed Description . . . . .	163
8.49.2 Property Documentation . . . . .	163
8.49.2.1 mode . . . . .	163
8.49.2.2 onOff . . . . .	163
8.49.2.3 parameter . . . . .	163
8.49.2.4 polarity . . . . .	163
8.49.2.5 source . . . . .	163
8.50 TriggerModelInfo Struct Reference . . . . .	164
8.50.1 Detailed Description . . . . .	164
8.50.2 Property Documentation . . . . .	164
8.50.2.1 modeMask . . . . .	164
8.50.2.2 onOffSupported . . . . .	164
8.50.2.3 polaritySupported . . . . .	164
8.50.2.4 present . . . . .	165
8.50.2.5 readOutSupported . . . . .	165
8.50.2.6 softwareTriggerSupported . . . . .	165
8.50.2.7 sourceMask . . . . .	165
8.50.2.8 valueReadable . . . . .	165

## **Chapter 1**

# **Software Licensing Information**

Component	License
<a href="#">FlyCapture2</a>	Copyright © 2017 FLIR Integrated Imaging Solutions, Inc. All Rights Reserved. This software is the confidential and proprietary information of FLIR Integrated Imaging Solutions, Inc. ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with FLIR Integrated Imaging Solutions, Inc. (FLIR). FLIR MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FLIR SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.
AdapterList	The Code Project Open License (CPOL) <a href="http://www.codeproject.com/info/cpol10.aspx">http://www.codeproject.com/info/cpol10.aspx</a>
Boost	Boost Software License <a href="http://www.boost.org/users/license.html">http://www.boost.org/users/license.html</a>
FFMPEG	LGPLv2.1 License <a href="https://www.ffmpeg.org/legal.html">https://www.ffmpeg.org/legal.html</a>
FreeImage	FreeImage public license <a href="http://freeimage.sourceforge.net/freeimage-license.txt">http://freeimage.sourceforge.net/freeimage-license.txt</a>
GTK	LGPLv2.1 License <a href="http://www.gnu.org/licenses/old-licenses/lgpl-2.1.txt">http://www.gnu.org/licenses/old-licenses/lgpl-2.1.txt</a>
Libusb	LGPLv2.1 License <a href="http://www.gnu.org/licenses/old-licenses/lgpl-2.1.txt">http://www.gnu.org/licenses/old-licenses/lgpl-2.1.txt</a>
Libraw1394	LGPLv2.0 License <a href="http://www.gnu.org/licenses/old-licenses/lgpl-2.0.txt">http://www.gnu.org/licenses/old-licenses/lgpl-2.0.txt</a>
Generated on Wed Apr 3 2019 19:09:01 for FlyCapture2 Managed by Doxygen	

Table 1.1: License table

# Chapter 2

## Module Index

### 2.1 Modules

Here is a list of all modules:

Enumerations . . . . .	13
Structures . . . . .	26
Image saving structures. . . . .	28





# Chapter 3

## Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">FlyCapture2</a> . . . . .	<a href="#">29</a>
<a href="#">FlyCapture2Managed</a> . . . . .	<a href="#">29</a>
<a href="#">FlyCapture2Managed::Gui</a> . . . . .	<a href="#">35</a>



## Chapter 4

# Class Index

### 4.1 Class Hierarchy

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

BMPOption . . . . .	37
CameraControlDialog . . . . .	37
CameraInfo . . . . .	38
CameraProperty . . . . .	43
CameraPropertyInfo . . . . .	45
CameraSelectionDialog . . . . .	48
CameraStats . . . . .	49
ConfigROM . . . . .	51
EmbeddedImageInfo . . . . .	53
EmbeddedImageInfoProperty . . . . .	54
FC2Config . . . . .	54
FC2Exception . . . . .	57
FC2Version . . . . .	59
Format7ImageSettings . . . . .	59
Format7Info . . . . .	61
Format7PacketInfo . . . . .	63
GigEConfig . . . . .	64
GigEImageSettings . . . . .	64
GigEImageSettingsInfo . . . . .	65
GigEProperty . . . . .	67
GigEStreamChannel . . . . .	68
ImageMetadata . . . . .	69
JpegOption . . . . .	71
Jpg2Option . . . . .	72
LutData . . . . .	73
ManagedBusManager . . . . .	74
ManagedCameraBase . . . . .	90
ManagedCamera . . . . .	83
ManagedGCCamera . . . . .	114

ManagedGigECamera . . . . .	117
ManagedEventCallbackData . . . . .	113
ManagedEventOptions . . . . .	114
ManagedGCPort . . . . .	116
ManagedImage . . . . .	126
ManagedImageStatistics . . . . .	136
ManagedPGRGuid . . . . .	137
ManagedTopologyNode . . . . .	139
ManagedUtilities . . . . .	143
NativeEventStruct . . . . .	144
PgmOption . . . . .	144
PngOption . . . . .	145
PpmOption . . . . .	146
StrobeControl . . . . .	147
StrobeInfo . . . . .	148
SystemInfo . . . . .	149
TiffOption . . . . .	151
TimeStamp . . . . .	152
Translate . . . . .	153
TriggerMode . . . . .	163
TriggerModelInfo . . . . .	164

## Chapter 5

# Class Index

### 5.1 Class List

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

<a href="#">BMPOption</a>	
Options for saving Bitmap image . . . . .	37
<a href="#">CameraControlDialog</a>	
<a href="#">CameraControlDialog</a> : managed wrapper of FlyCapture2::Camera- ControlDialog (see for details) . . . . .	37
<a href="#">CameraInfo</a>	
Camera information . . . . .	38
<a href="#">CameraProperty</a>	
A specific camera property . . . . .	43
<a href="#">CameraPropertyInfo</a>	
Information about a specific camera property . . . . .	45
<a href="#">CameraSelectionDialog</a>	
<a href="#">CameraControlDialog</a> : managed wrapper of FlyCapture2::Camera- SelectionDialog (see for details) . . . . .	48
<a href="#">CameraStats</a>	
Camera diagnostic information . . . . .	49
<a href="#">ConfigROM</a>	
Camera configuration ROM . . . . .	51
<a href="#">EmbeddedImageInfo</a>	
Properties of the possible embedded image information . . . . .	53
<a href="#">EmbeddedImageInfoProperty</a>	
Properties of a single embedded image info property . . . . .	54
<a href="#">FC2Config</a>	
Configuration for a camera . . . . .	54
<a href="#">FC2Exception</a>	
Exception that is thrown when an error is encountered . . . . .	57
<a href="#">FC2Version</a>	
The current version of the library . . . . .	59

<a href="#">Format7ImageSettings</a>	
Format 7 image settings . . . . .	59
<a href="#">Format7Info</a>	
Format 7 information for a single mode . . . . .	61
<a href="#">Format7PacketInfo</a>	
Format 7 packet information . . . . .	63
<a href="#">GigEConfig</a>	
Configuration for a GigE camera . . . . .	64
<a href="#">GigEImageSettings</a>	
Image settings for a GigE camera . . . . .	64
<a href="#">GigEImageSettingsInfo</a>	
Format 7 information for a single mode . . . . .	65
<a href="#">GigEProperty</a>	
A GigE property . . . . .	67
<a href="#">GigEStreamChannel</a>	
Information about a single GigE stream channel . . . . .	68
<a href="#">ImageMetadata</a>	
Metadata related to an image . . . . .	69
<a href="#">JpegOption</a>	
Options for saving JPEG image . . . . .	71
<a href="#">Jpg2Option</a>	
Options for saving JPEG2000 image . . . . .	72
<a href="#">LutData</a>	
Information about the camera's look up table . . . . .	73
<a href="#">ManagedBusManager</a>	
<a href="#">ManagedBusManager</a> provides the functionality for the user to get an PGRGuid for a desired camera or device easily . . . . .	74
<a href="#">ManagedCamera</a>	
<a href="#">ManagedCamera</a> represents a physical camera that uses the IIDC register set . . . . .	83
<a href="#">ManagedCameraBase</a>	
Abstract base class that represents a generic camera that defines a general interface to a camera . . . . .	90
<a href="#">ManagedEventCallbackData</a> . . . . .	113
<a href="#">ManagedEventOptions</a>	
Options for enabling device event registration . . . . .	114
<a href="#">ManagedGCCamera</a> . . . . .	114
<a href="#">ManagedGCPort</a> . . . . .	116
<a href="#">ManagedGigECamera</a>	
The GigECamera object represents a physical Gigabit Ethernet cam- era . . . . .	117
<a href="#">ManagedImage</a>	
The ManagedImage class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk . . . . .	126
<a href="#">ManagedImageStatistics</a> . . . . .	136
<a href="#">ManagedPGRGuid</a>	
Managed version of a PGRGuid . . . . .	137

<a href="#">ManagedTopologyNode</a>	
Topology information that can be used to generate a tree structure of all cameras and devices connected to a computer . . . . .	139
<a href="#">ManagedUtilities</a> . . . . .	143
<a href="#">NativeEventStruct</a> . . . . .	144
<a href="#">PgmOption</a>	
Options for saving PGM images . . . . .	144
<a href="#">PngOption</a>	
Options for saving PNG images . . . . .	145
<a href="#">PpmOption</a>	
Options for saving PPM images . . . . .	146
<a href="#">StrobeControl</a>	
A camera strobe . . . . .	147
<a href="#">StrobeInfo</a>	
A camera strobe property . . . . .	148
<a href="#">SystemInfo</a>	
Description of the system . . . . .	149
<a href="#">TiffOption</a>	
Options for saving TIFF images . . . . .	151
<a href="#">TimeStamp</a>	
Timestamp information . . . . .	152
<a href="#">Translate</a> . . . . .	153
<a href="#">TriggerMode</a>	
A camera trigger . . . . .	163
<a href="#">TriggerModelInfo</a>	
Information about a camera trigger property . . . . .	164





## Chapter 6

# Module Documentation

### 6.1 Enumerations

#### Enumerations

- enum `ErrorType` { `Undefined` = -1, `Ok`, `Failed`, `NotImplemented`, `FailedBusMasterConnection`, `NotConnected`, `InitFailed`, `NotInitialized`, `InvalidParameter`, `InvalidSettings`, `InvalidBuManager`, `MemoryAllocationFailed`, `LowLevelFailure`, `NotFound`, `FailedGuid`, `InvalidPacketSize`, `InvalidMode`, `NotInFormat7`, `×` `NotSupported`, `Timeout`, `BusMasterFailed`, `InvalidGeneration`, `LutFailed`, `×` `lfdcFailed`, `StrobeFailed`, `TriggerFailed`, `PropertyFailed`, `PropertyNotPresent`, `RegisterFailed`, `ReadRegisterFailed`, `WriteRegisterFailed`, `IsochFailed`, `×` `IsochAlreadyStarted`, `IsochNotStarted`, `IsochStartFailed`, `IsochRetrieveBufferFailed`, `IsochStopFailed`, `IsochSyncFailed`, `IsochBandwidthExceeded`, `ImageConversionFailed`, `ImageLibraryFailure`, `BufferTooSmall`, `ImageConsistencyError`, `IncompatibleDriver` }

*The error types returned by functions.*

- enum `ManagedCallbackType` { `BusReset`, `Arrival`, `Removal` }

*The type of bus callback to register a callback function for.*

- enum `GrabMode` { `DropFrames`, `BufferFrames`, `Unspecified` = -2 }

*The grab strategy employed during image transfer.*

- enum `GrabTimeout` { `None` = 0, `Infinite` = -1, `Unspecified` = -2 }

*Timeout options for grabbing images.*

- enum `BandwidthAllocation` { `Off` = 0, `On` = 1, `Unsupported` = 2, `Unspecified` = -2 }

*Bandwidth allocation options for 1394 devices.*

- enum `InterfaceType` { `Ieee1394`, `Usb2`, `Usb3`, `GigE`, `Unknown` = -1 }

*Interfaces that a camera may use to communicate with a host.*

- enum `PropertyType` { `Brightness`, `AutoExposure`, `Sharpness`, `WhiteBalance`, `Hue`, `Saturation`, `Gamma`, `Iris`, `Focus`, `Zoom`, `Pan`, `Tilt`, `Shutter`, `Gain`, `TriggerMode`, `TriggerDelay`, `FrameRate`, `Temperature`, `Unspecified` = -2 }

*Camera properties.*

- enum `FrameRate` { `FrameRate1_875`, `FrameRate3_75`, `FrameRate7_5`, `FrameRate15`, `FrameRate30`, `FrameRate60`, `FrameRate120`, `FrameRate240`, `FrameRateFormat7`, `NumberOfFrameRates` }

*Frame rates in frames per second.*

- enum `VideoMode` { `VideoMode160x120Yuv444`, `VideoMode320x240Yuv422`, `VideoMode640x480Yuv411`, `VideoMode640x480Yuv422`, `VideoMode640x480Rbg`, `VideoMode640x480Y8`, `VideoMode640x480Y16`, `VideoMode800x600Yuv422`, `VideoMode800x600Rbg`, `VideoMode800x600Y8`, `VideoMode800x600Y16`, `VideoMode1024x768Yuv422`, `VideoMode1024x768Rbg`, `VideoMode1024x768Y8`, `VideoMode1024x768Y16`, `VideoMode1280x960Yuv422`, `VideoMode1280x960Rbg`, `VideoMode1280x960Y8`, `VideoMode1280x960Y16`, `VideoMode1600x1200Yuv422`, `VideoMode1600x1200Rbg`, `VideoMode1600x1200Y8`, `VideoMode1600x1200Y16`, `VideoModeFormat7`, `NumberOfVideoModes` }

*DCAM video modes.*

- enum `Mode` { `Mode0` = 0, `Mode1`, `Mode2`, `Mode3`, `Mode4`, `Mode5`, `Mode6`, `Mode7`, `Mode8`, `Mode9`, `Mode10`, `Mode11`, `Mode12`, `Mode13`, `Mode14`, `Mode15`, `Mode16`, `Mode17`, `Mode18`, `Mode19`, `Mode20`, `Mode21`, `Mode22`, `Mode23`, `Mode24`, `Mode25`, `Mode26`, `Mode27`, `Mode28`, `Mode29`, `Mode30`, `Mode31`, `NumberOfModes` }

*Camera modes for DCAM formats as well as Format7.*

- enum `PixelFormat` { `PixelFormatMono8` = 0x80000000, `PixelFormat411Yuv8` = 0x40000000, `PixelFormat422Yuv8` = 0x20000000, `PixelFormat444Yuv8` = 0x10000000, `PixelFormatRgb8` = 0x08000000, `PixelFormatMono16` = 0x04000000, `PixelFormatRgb16` = 0x02000000, `PixelFormatSignedMono16` = 0x01000000, `PixelFormatSignedRgb16` = 0x00800000, `PixelFormatRaw8` = 0x00400000, `PixelFormatRaw16` = 0x00200000, `PixelFormatMono12` = 0x00100000, `PixelFormatRaw12` = 0x00080000, `PixelFormatBgr` = 0x80000008, `PixelFormatBgru` = 0x40000008, `PixelFormatRgb` = `PixelFormatRgb8`, `PixelFormatRgbu` = 0x40000002, `PixelFormatBgr16` = 0x02000001, `PixelFormatBgru16` = 0x02000002, `PixelFormat422Yuv8Jpeg` = 0x40000001, `NumberOfPixelFormats` = 20 }

*Pixel formats available for Format7 modes.*

- enum `BusSpeed` { `S100`, `S200`, `S400`, `S480`, `S800`, `S1600`, `S3200`, `S5000`, `GigE_10Base_T`, `GigE_100Base_T`, `GigE_1000Base_T`, `GigE_10000Base_T`, `Fastest`, `Any`, `Unknown` = -1 }

*Bus speeds.*

- enum `PCleBusSpeed` { `Speed_2_5`, `Speed_5_0`, `Unknown` = -1 }

*PCle Bus Speeds.*

- enum `DriverType` { `leee1394_Cam`, `leee1394_Pro`, `leee1394_Juju`, `leee1394_Video1394`, `leee1394_Raw1394`, `Usb_None`, `Usb_Cam`, `Usb3_Pro`, `GigE_None`, `GigE_Filter`, `GigE_Pro`, `GigE_Lwf`, `Unknown` = -1 }

*Types of low level drivers that flycapture uses.*

- enum `ColorProcessingAlgorithm` { `Default`, `NoColorProcessing`, `NearestNeighbor`, `EdgeSensing`, `HQLinear`, `Rigorous`, `IPP`, `Directional`, `WeightedDirectional` }

*Color processing algorithms.*

- enum `BayerTileFormat` { `None` = 0, `RGBG`, `GRBG`, `GBRG`, `BGGR` }  
*Bayer tile formats.*
- enum `ImageFileFormat` { `FromFileExtension` = -1, `Pgm`, `Ppm`, `Bmp`, `Jpeg`, `Jpeg2000`, `Tiff`, `Png`, `Raw` }  
*File formats to be used for saving images to disk.*
- enum `GigEPropertyType` { `Heartbeat`, `HeartbeatTimeout`, `PacketSize`, `PacketDelay` }  
*Possible properties that can be queried from the camera.*
- enum `StatisticsChannel` { `Grey`, `Red`, `Green`, `Blue`, `Hue`, `Saturation`, `Lightness`, `NumberOfStatisticsChannels` }  
*Channels that allow statistics to be calculated.*
- enum `OSType` { `WindowsX86`, `WindowsX64`, `LinuxX86`, `LinuxX64`, `Mac`, `UnknownOS` }  
*Possible operating systems.*
- enum `ByteOrder` { `LittleEndian`, `BigEndian` }  
*Possible byte orders.*

### 6.1.1 Enumeration Type Documentation

#### 6.1.1.1 enum BandwidthAllocation

Bandwidth allocation options for 1394 devices.

Enumerator:

- Off** Do not allocate bandwidth.
- On** Allocate bandwidth. This is the default setting.
- Unsupported** Bandwidth allocation is not supported by either the camera or operating system.
- Unspecified** Unspecified grab mode. Unspecified property type.  
Not specified.  
Unspecified timeout setting.  
This leaves the current setting unchanged.

#### 6.1.1.2 enum BayerTileFormat

Bayer tile formats.

Enumerator:

- None** Non-blocking wait. No bayer tile format.
- RGGB** Red-Green-Green-Blue.
- GRBG** Green-Red-Blue-Green.
- GBRG** Green-Blue-Red-Green.
- BGGR** Blue-Green-Green-Red.

#### 6.1.1.3 enum BusSpeed

Bus speeds.

Enumerator:

- S100** 100Mbps/sec.
- S200** 200Mbps/sec.
- S400** 400Mbps/sec.
- S480** 480Mbps/sec. Only for USB2 cameras.
- S800** 800Mbps/sec.
- S1600** 1600Mbps/sec.
- S3200** 3200Mbps/sec.
- S5000** 5000Mbps/sec. Only for USB3 cameras.
- GigE\_10Base\_T**
- GigE\_100Base\_T**
- GigE\_1000Base\_T**
- GigE\_10000Base\_T**
- Fastest** The fastest speed available.
- Any** Any speed that is available.
- Unknown** Unknown interface. Unknown driver type.
  - 5.0 Gb/s
  - Unknown bus speed.
  - Speed is unknown

#### 6.1.1.4 enum ByteOrder

Possible byte orders.

Enumerator:

- LittleEndian**
- BigEndian**

#### 6.1.1.5 enum ColorProcessingAlgorithm

Color processing algorithms.

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

Enumerator:

- Default** Default method.
- NoColorProcessing** No color processing.
- NearestNeighbor** Fastest but lowest quality. Equivalent to FLYCAPTURE\_NEAREST\_NEIGHBOR\_FAST in FlyCapture.
- EdgeSensing** Weights surrounding pixels based on localized edge orientation.
- HQLinear** Similar quality to rigorous but much faster.
- Rigorous** Slowest but produces the best results.
- IPP** Multithreaded with similar results to edge sensing.
- Directional** Best quality but much faster than rigorous.
- WeightedDirectional** Weighted pixel average from different directions.

#### 6.1.1.6 enum DriverType

Types of low level drivers that flycapture uses.

Enumerator:

- lee1394\_Cam** PGRCam.sys.
- lee1394\_Pro** PGR1394.sys.
- lee1394\_Juju** firewire\_core.
- lee1394\_Video1394** video1394.
- lee1394\_Raw1394** raw1394.
- Usb\_None** No usb driver used just BSD stack. (Linux only)
- Usb\_Cam** PGRUsbCam.sys.
- Usb3\_Pro** PGRXHCl.sys.
- GigE\_None** no gige drivers used,MS/BSD stack.
- GigE\_Filter** PGRGigE.sys.
- GigE\_Pro** PGRGigEPro.sys.
- GigE\_Lwf** PgrLwf.sys.
- Unknown** Unknown interface. Unknown driver type.
  - 5.0 Gb/s
  - Unknown bus speed.
  - Speed is unknown

#### 6.1.1.7 enum ErrorType

The error types returned by functions.

Enumerator:

- Undefined** Undefined.

**Ok** Function returned with no errors.

**Failed** General failure.

**NotImplemented** Function has not been implemented.

**FailedBusMasterConnection** Could not connect to Bus Master.

**NotConnected** Camera has not been connected.

**InitFailed** Initialization failed.

**NotInitialized** Camera has not been initialized.

**InvalidParameter** Invalid parameter passed to function.

**InvalidSettings** Setting set to camera is invalid.

**InvalidBuManager** Invalid Bus Manager object.

**MemoryAllocationFailed** Could not allocate memory.

**LowLevelFailure** Low level error.

**NotFound** Device not found.

**FailedGuid** GUID failure.

**InvalidPacketSize** Packet size set to camera is invalid.

**InvalidMode** Invalid mode has been passed to function.

**NotInFormat7** Error due to not being in Format7.

**NotSupported** This feature is unsupported.

**Timeout** Timeout error.

**BusMasterFailed** Bus Master Failure.

**InvalidGeneration** Generation Count Mismatch.

**LutFailed** Look Up Table failure.

**IidcFailed** IIDC failure.

**StrobeFailed** Strobe failure.

**TriggerFailed** Trigger failure.

**PropertyFailed** Property failure.

**PropertyNotPresent** Property is not present.

**RegisterFailed** Register access failed.

**ReadRegisterFailed** Register read failed.

**WriteRegisterFailed** Register write failed.

**IsochFailed** Isochronous failure.

**IsochAlreadyStarted** Isochronous transfer has already been started.

**IsochNotStarted** Isochronous transfer has not been started.

**IsochStartFailed** Isochronous start failed.

**IsochRetrieveBufferFailed** Isochronous retrieve buffer failed.

**IsochStopFailed** Isochronous stop failed.

**IsochSyncFailed** Isochronous image synchronization failed.

**IsochBandwidthExceeded** Isochronous bandwidth exceeded.

***ImageConversionFailed*** Image conversion failed.

***ImageLibraryFailure*** Image library failure.

***BufferTooSmall*** Buffer is too small.

***ImageConsistencyError*** There is an image consistency error.

***IncompatibleDriver*** IncompatibleDriver error.

#### 6.1.1.8 enum FrameRate

Frame rates in frames per second.

Enumerator:

***FrameRate1\_875*** 1.875 fps.

***FrameRate3\_75*** 3.75 fps.

***FrameRate7\_5*** 7.5 fps.

***FrameRate15*** 15 fps.

***FrameRate30*** 30 fps.

***FrameRate60*** 60 fps.

***FrameRate120*** 120 fps.

***FrameRate240*** 240 fps.

***FrameRateFormat7*** Custom frame rate for Format7 functionality.

***NumberOfFrameRates*** Number of possible camera frame rates.

#### 6.1.1.9 enum GigEPropertyType

Possible properties that can be queried from the camera.

Enumerator:

***Heartbeat***

***HeartbeatTimeout***

***PacketSize***

***PacketDelay***

#### 6.1.1.10 enum GrabMode

The grab strategy employed during image transfer.

This type controls how images that stream off the camera accumulate in a user buffer for handling. Unlike earlier versions of the FlyCapture SDK, it is no longer necessary to explicitly start the image grabbing process before specifying an image grabbing mode.

Enumerator:

**DropFrames** Grabs the newest image in the user buffer each time the Retrieve-Buffer() function is called. Older images are dropped instead of accumulating in the user buffer. Grabbing blocks if the camera has not finished transmitting the next available image. If the camera is transmitting images faster than the application can grab them, images may be dropped and only the most recent image is stored for grabbing. Note that this mode is the equivalent of flycaptureLockLatest in earlier versions of the FlyCapture SDK.

**BufferFrames** Images accumulate in the user buffer, and the oldest image is grabbed for handling before being discarded. This member can be used to guarantee that each image is seen. However, image processing time must not exceed transmission time from the camera to the buffer. Grabbing blocks if the camera has not finished transmitting the next available image. The buffer size is controlled by the numBuffers parameter in the [FC2Config](#) struct. Note that this mode is the equivalent of flycaptureLockNext in earlier versions of the FlyCapture SDK.

**Unspecified** Unspecified grab mode. Unspecified property type.  
 Not specified.  
 Unspecified timeout setting.  
 This leaves the current setting unchanged.

#### 6.1.1.11 enum GrabTimeout

Timeout options for grabbing images.

Enumerator:

**None** Non-blocking wait. No bayer tile format.  
**Infinite** Wait indefinitely.  
**Unspecified** Unspecified grab mode. Unspecified property type.  
 Not specified.  
 Unspecified timeout setting.  
 This leaves the current setting unchanged.

#### 6.1.1.12 enum ImageFileFormat

File formats to be used for saving images to disk.

Enumerator:

**FromFileExtension** Determine file format from file extension.  
**Pgm** Portable gray map.  
**Ppm** Portable pixmap.  
**Bmp** Bitmap.



**Jpeg** JPEG.  
**Jpeg2000** JPEG 2000.  
**Tiff** Tagged image file format.  
**Png** Portable network graphics.  
**Raw** Raw data.

#### 6.1.1.13 enum InterfaceType

Interfaces that a camera may use to communicate with a host.

Enumerator:

**ieee1394** IEEE-1394 (Includes 1394a and 1394b).  
**Usb2** USB 2.0.  
**Usb3** USB 3.0.  
**GigE** GigE.  
**Unknown** Unknown interface. Unknown driver type.  
5.0 Gb/s  
Unknown bus speed.  
Speed is unknown

#### 6.1.1.14 enum ManagedCallbackType

The type of bus callback to register a callback function for.

Enumerator:

**BusReset** Register for all bus events.  
**Arrival** Register for arrivals only.  
**Removal** Register for removals only.

#### 6.1.1.15 enum Mode

Camera modes for DCAM formats as well as Format7.

Enumerator:

**Mode0**  
**Mode1**  
**Mode2**  
**Mode3**  
**Mode4**

**Mode5**  
**Mode6**  
**Mode7**  
**Mode8**  
**Mode9**  
**Mode10**  
**Mode11**  
**Mode12**  
**Mode13**  
**Mode14**  
**Mode15**  
**Mode16**  
**Mode17**  
**Mode18**  
**Mode19**  
**Mode20**  
**Mode21**  
**Mode22**  
**Mode23**  
**Mode24**  
**Mode25**  
**Mode26**  
**Mode27**  
**Mode28**  
**Mode29**  
**Mode30**  
**Mode31**  
**NumberOfModes**

#### 6.1.1.16 enum OSType

Possible operating systems.

Enumerator:

**WindowsX86** All Windows 32-bit variants.  
**WindowsX64** All Windows 64-bit variants.  
**LinuxX86** All Linux 32-bit variants.  
**LinuxX64** All Linux 32-bit variants.  
**Mac** Mac OSX.  
**UnknownOS** Unknown operating system.

## 6.1.1.17 enum PCIeBusSpeed

PCIe Bus Speeds.

Enumerator:

**Speed\_2\_5**

**Speed\_5\_0** 2.5 Gb/s

**Unknown** Unknown interface. Unknown driver type.

5.0 Gb/s

Unknown bus speed.

Speed is unknown

## 6.1.1.18 enum PixelFormat

Pixel formats available for Format7 modes.

Enumerator:

**PixelFormatMono8** 8 bits of mono information.

**PixelFormat411Yuv8** YUV 4:1:1.

**PixelFormat422Yuv8** YUV 4:2:2.

**PixelFormat444Yuv8** YUV 4:4:4.

**PixelFormatRgb8** R = G = B = 8 bits.

**PixelFormatMono16** 16 bits of mono information.

**PixelFormatRgb16** R = G = B = 16 bits.

**PixelFormatSignedMono16** 16 bits of signed mono information.

**PixelFormatSignedRgb16** R = G = B = 16 bits signed.

**PixelFormatRaw8** 8 bit raw data output of sensor.

**PixelFormatRaw16** 16 bit raw data output of sensor.

**PixelFormatMono12** 12 bits of mono information.

**PixelFormatRaw12** 12 bit raw data output of sensor.

**PixelFormatBgr** 24 bit BGR.

**PixelFormatBgru** 32 bit BGRU.

**PixelFormatRgb** 24 bit RGB.

**PixelFormatRgbu** 32 bit RGBU.

**PixelFormatBgr16** R = G = B = 16 bits.

**PixelFormatBgru16** 64 bit BGRU.

**PixelFormat422Yuv8Jpeg** JPEG compressed stream.

**NumberOfPixelFormat** Number of pixel formats.

#### 6.1.1.19 enum PropertyType

Camera properties.

Not all properties may be supported, depending on the camera model.

Enumerator:

**Brightness** Brightness.  
**AutoExposure** Auto exposure.  
**Sharpness** Sharpness.  
**WhiteBalance** White balance.  
**Hue** Hue.  
**Saturation** Saturation.  
**Gamma** Gamma.  
**Iris** Iris.  
**Focus** Focus.  
**Zoom** Zoom.  
**Pan** Pan.  
**Tilt** Tilt.  
**Shutter** Shutter.  
**Gain** Gain.  
**TriggerMode** Trigger mode.  
**TriggerDelay** Trigger delay.  
**FrameRate** Frame rate.  
**Temperature** Temperature.  
**Unspecified** Unspecified grab mode. Unspecified property type.  
Not specified.  
Unspecified timeout setting.  
This leaves the current setting unchanged.

#### 6.1.1.20 enum StatisticsChannel

Channels that allow statistics to be calculated.

Enumerator:

**Grey**  
**Red**  
**Green**  
**Blue**  
**Hue** Hue.  
**Saturation** Saturation.  
**Lightness**  
**NumberOfStatisticsChannels**

## 6.1.1.21 enum VideoMode

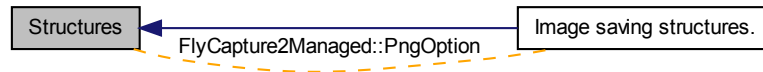
DCAM video modes.

Enumerator:

**VideoMode160x120Yuv444** 160x120 YUV444.  
**VideoMode320x240Yuv422** 320x240 YUV422.  
**VideoMode640x480Yuv411** 640x480 YUV411.  
**VideoMode640x480Yuv422** 640x480 YUV422.  
**VideoMode640x480Rgb** 640x480 24-bit RGB.  
**VideoMode640x480Y8** 640x480 8-bit.  
**VideoMode640x480Y16** 640x480 16-bit.  
**VideoMode800x600Yuv422** 800x600 YUV422.  
**VideoMode800x600Rgb** 800x600 RGB.  
**VideoMode800x600Y8** 800x600 8-bit.  
**VideoMode800x600Y16** 800x600 16-bit.  
**VideoMode1024x768Yuv422** 1024x768 YUV422.  
**VideoMode1024x768Rgb** 1024x768 RGB.  
**VideoMode1024x768Y8** 1024x768 8-bit.  
**VideoMode1024x768Y16** 1024x768 16-bit.  
**VideoMode1280x960Yuv422** 1280x960 YUV422.  
**VideoMode1280x960Rgb** 1280x960 RGB.  
**VideoMode1280x960Y8** 1280x960 8-bit.  
**VideoMode1280x960Y16** 1280x960 16-bit.  
**VideoMode1600x1200Yuv422** 1600x1200 YUV422.  
**VideoMode1600x1200Rgb** 1600x1200 RGB.  
**VideoMode1600x1200Y8** 1600x1200 8-bit.  
**VideoMode1600x1200Y16** 1600x1200 16-bit.  
**VideoModeFormat7** Custom video mode for Format7 functionality.  
**NumberOfVideoModes** Number of possible video modes.

## 6.2 Structures

Collaboration diagram for Structures:



### Classes

- struct [FC2Version](#)  
*The current version of the library.*
- struct [GigEProperty](#)  
*A GigE property.*
- struct [GigEStreamChannel](#)  
*Information about a single GigE stream channel.*
- struct [GigEConfig](#)  
*Configuration for a GigE camera.*
- struct [GigEImageSettingsInfo](#)  
*Format 7 information for a single mode.*
- struct [GigEImageSettings](#)  
*Image settings for a GigE camera.*
- struct [FC2Config](#)  
*Configuration for a camera.*
- struct [CameraPropertyInfo](#)  
*Information about a specific camera property.*
- struct [CameraProperty](#)  
*A specific camera property.*
- struct [TriggerModelInfo](#)  
*Information about a camera trigger property.*
- struct [TriggerMode](#)  
*A camera trigger.*
- struct [StrobeInfo](#)  
*A camera strobe property.*
- struct [StrobeControl](#)  
*A camera strobe.*
- struct [Format7ImageSettings](#)  
*Format 7 image settings.*
- struct [Format7Info](#)

- Format 7 information for a single mode.*
  - struct [Format7PacketInfo](#)  
*Format 7 packet information.*
- struct [TimeStamp](#)  
*Timestamp information.*
- struct [ConfigROM](#)  
*Camera configuration ROM.*
- struct [CameraInfo](#)  
*Camera information.*
- struct [EmbeddedImageInfoProperty](#)  
*Properties of a single embedded image info property.*
- struct [EmbeddedImageInfo](#)  
*Properties of the possible embedded image information.*
- struct [ImageMetadata](#)  
*Metadata related to an image.*
- struct [LutData](#)  
*Information about the camera's look up table.*
- struct [CameraStats](#)  
*Camera diagnostic information.*
- struct [PngOption](#)  
*Options for saving PNG images.*

## Modules

- [Image saving structures.](#)  
*These structures define various parameters used for saving images.*

## 6.3 Image saving structures.

These structures define various parameters used for saving images.

Collaboration diagram for Image saving structures.:



### Classes

- struct [PngOption](#)  
*Options for saving PNG images.*
- struct [PpmOption](#)  
*Options for saving PPM images.*
- struct [PgmOption](#)  
*Options for saving PGM images.*
- struct [TiffOption](#)  
*Options for saving TIFF images.*
- struct [JpegOption](#)  
*Options for saving JPEG image.*
- struct [Jpg2Option](#)  
*Options for saving JPEG2000 image.*
- struct [BMPOption](#)  
*Options for saving Bitmap image.*
- struct [SystemInfo](#)  
*Description of the system.*

### 6.3.1 Detailed Description

These structures define various parameters used for saving images.



## Chapter 7

# Namespace Documentation

### 7.1 FlyCapture2 Namespace Reference

### 7.2 FlyCapture2Managed Namespace Reference

#### Namespaces

- namespace [Gui](#)

#### Classes

- class [FC2Exception](#)  
*Exception that is thrown when an error is encountered.*
- class [ManagedBusManager](#)  
*[ManagedBusManager](#) provides the functionality for the user to get an PGRGuid for a desired camera or device easily.*
- class [ManagedCamera](#)  
*[ManagedCamera](#) represents a physical camera that uses the IIDC register set.*
- struct [ManagedEventCallbackData](#)
- struct [ManagedEventOptions](#)  
*Options for enabling device event registration.*
- struct [NativeEventStruct](#)
- class [ManagedCameraBase](#)  
*Abstract base class that represents a generic camera that defines a general interface to a camera.*
- struct [FC2Version](#)  
*The current version of the library.*
- struct [GigEProperty](#)  
*A GigE property.*
- struct [GigEStreamChannel](#)

*Information about a single GigE stream channel.*

- struct [GigEConfig](#)

*Configuration for a GigE camera.*

- struct [GigEImageSettingsInfo](#)

*Format 7 information for a single mode.*

- struct [GigEImageSettings](#)

*Image settings for a GigE camera.*

- struct [FC2Config](#)

*Configuration for a camera.*

- struct [CameraPropertyInfo](#)

*Information about a specific camera property.*

- struct [CameraProperty](#)

*A specific camera property.*

- struct [TriggerModelInfo](#)

*Information about a camera trigger property.*

- struct [TriggerMode](#)

*A camera trigger.*

- struct [StrobeInfo](#)

*A camera strobe property.*

- struct [StrobeControl](#)

*A camera strobe.*

- struct [Format7ImageSettings](#)

*Format 7 image settings.*

- struct [Format7Info](#)

*Format 7 information for a single mode.*

- struct [Format7PacketInfo](#)

*Format 7 packet information.*

- struct [TimeStamp](#)

*Timestamp information.*

- struct [ConfigROM](#)

*Camera configuration ROM.*

- struct [CameraInfo](#)

*Camera information.*

- struct [EmbeddedImageInfoProperty](#)

*Properties of a single embedded image info property.*

- struct [EmbeddedImageInfo](#)

*Properties of the possible embedded image information.*

- struct [ImageMetadata](#)

*Metadata related to an image.*

- struct [LutData](#)

*Information about the camera's look up table.*

- struct [CameraStats](#)

*Camera diagnostic information.*

- struct [PngOption](#)  
*Options for saving PNG images.*
- struct [PpmOption](#)  
*Options for saving PPM images.*
- struct [PgmOption](#)  
*Options for saving PGM images.*
- struct [TiffOption](#)  
*Options for saving TIFF images.*
- struct [JpegOption](#)  
*Options for saving JPEG image.*
- struct [Jpg2Option](#)  
*Options for saving JPEG2000 image.*
- struct [BMPOption](#)  
*Options for saving Bitmap image.*
- struct [SystemInfo](#)  
*Description of the system.*
- class [ManagedGCCamera](#)
- class [ManagedGCPort](#)
- class [ManagedGigECamera](#)  
*The GigECamera object represents a physical Gigabit Ethernet camera.*
- class [ManagedImage](#)  
*The ManagedImage class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk.*
- class [ManagedImageStatistics](#)
- class [ManagedPGRGuid](#)  
*Managed version of a PGRGuid.*
- class [ManagedTopologyNode](#)  
*The ManagedTopologyNode class contains topology information that can be used to generate a tree structure of all cameras and devices connected to a computer.*
- class [ManagedUtilities](#)
- class [Translate](#)

## Enumerations

- enum [ErrorType](#) { [Undefined](#) = -1, [Ok](#), [Failed](#), [NotImplemented](#), [FailedBusMasterConnection](#), [NotConnected](#), [InitFailed](#), [NotInitialized](#), [InvalidParameter](#), [InvalidSettings](#), [InvalidBuManager](#), [MemoryAllocationFailed](#), [LowLevelFailure](#), [NotFound](#), [FailedGuid](#), [InvalidPacketSize](#), [InvalidMode](#), [NotInFormat7](#), × [NotSupported](#), [Timeout](#), [BusMasterFailed](#), [InvalidGeneration](#), [LutFailed](#), × [IldcFailed](#), [StrobeFailed](#), [TriggerFailed](#), [PropertyFailed](#), [PropertyNotPresent](#), [RegisterFailed](#), [ReadRegisterFailed](#), [WriteRegisterFailed](#), [IsochFailed](#), × [IsochAlreadyStarted](#), [IsochNotStarted](#), [IsochStartFailed](#), [IsochRetrieveBufferFailed](#), [IsochStopFailed](#), [IsochSyncFailed](#), [IsochBandwidthExceeded](#), [ImageConversionFailed](#), [ImageLibraryFailure](#), [BufferTooSmall](#), [ImageConsistencyError](#), [IncompatibleDriver](#) }

*The error types returned by functions.*

- enum [ManagedCallbackType](#) { [BusReset](#), [Arrival](#), [Removal](#) }

*The type of bus callback to register a callback function for.*

- enum [GrabMode](#) { [DropFrames](#), [BufferFrames](#), [Unspecified](#) = -2 }

*The grab strategy employed during image transfer.*

- enum [GrabTimeout](#) { [None](#) = 0, [Infinite](#) = -1, [Unspecified](#) = -2 }

*Timeout options for grabbing images.*

- enum [BandwidthAllocation](#) { [Off](#) = 0, [On](#) = 1, [Unsupported](#) = 2, [Unspecified](#) = -2 }

*Bandwidth allocation options for 1394 devices.*

- enum [InterfaceType](#) { [Ieee1394](#), [Usb2](#), [Usb3](#), [GigE](#), [Unknown](#) = -1 }

*Interfaces that a camera may use to communicate with a host.*

- enum [PropertyType](#) { [Brightness](#), [AutoExposure](#), [Sharpness](#), [WhiteBalance](#), [Hue](#), [Saturation](#), [Gamma](#), [Iris](#), [Focus](#), [Zoom](#), [Pan](#), [Tilt](#), [Shutter](#), [Gain](#), [TriggerMode](#), [TriggerDelay](#), [FrameRate](#), [Temperature](#), [Unspecified](#) = -2 }

*Camera properties.*

- enum [FrameRate](#) { [FrameRate1\\_875](#), [FrameRate3\\_75](#), [FrameRate7\\_5](#), [FrameRate15](#), [FrameRate30](#), [FrameRate60](#), [FrameRate120](#), [FrameRate240](#), [FrameRateFormat7](#), [NumberOfFrameRates](#) }

*Frame rates in frames per second.*

- enum [VideoMode](#) { [VideoMode160x120Yuv444](#), [VideoMode320x240Yuv422](#), [VideoMode640x480Yuv411](#), [VideoMode640x480Yuv422](#), [VideoMode640x480Rgb](#), [VideoMode640x480Y8](#), [VideoMode640x480Y16](#), [VideoMode800x600Yuv422](#), [VideoMode800x600Rgb](#), [VideoMode800x600Y8](#), [VideoMode800x600Y16](#), [VideoMode1024x768Yuv422](#), [VideoMode1024x768Rgb](#), [VideoMode1024x768Y8](#), [VideoMode1024x768Y16](#), [VideoMode1280x960Yuv422](#), [VideoMode1280x960Rgb](#), [VideoMode1280x960Y8](#), [VideoMode1280x960Y16](#), [VideoMode1600x1200Yuv422](#), [VideoMode1600x1200Rgb](#), [VideoMode1600x1200Y8](#), [VideoMode1600x1200Y16](#), [VideoModeFormat7](#), [NumberOfVideoModes](#) }

*DCAM video modes.*

- enum [Mode](#) { [Mode0](#) = 0, [Mode1](#), [Mode2](#), [Mode3](#), [Mode4](#), [Mode5](#), [Mode6](#), [Mode7](#), [Mode8](#), [Mode9](#), [Mode10](#), [Mode11](#), [Mode12](#), [Mode13](#), [Mode14](#), [Mode15](#), [Mode16](#), [Mode17](#), [Mode18](#), [Mode19](#), [Mode20](#), [Mode21](#), [Mode22](#), [Mode23](#), [Mode24](#), [Mode25](#), [Mode26](#), [Mode27](#), [Mode28](#), [Mode29](#), [Mode30](#), [Mode31](#), [NumberOfModes](#) }

*Camera modes for DCAM formats as well as Format7.*

- enum [PixelFormat](#) { [PixelFormatMono8](#) = 0x80000000, [PixelFormat411Yuv8](#) = 0x40000000, [PixelFormat422Yuv8](#) = 0x20000000, [PixelFormat444Yuv8](#) = 0x10000000, [PixelFormatRgb8](#) = 0x08000000, [PixelFormatMono16](#) = 0x04000000, [PixelFormatRgb16](#) = 0x02000000, [PixelFormatSignedMono16](#) = 0x01000000, [PixelFormatSignedRgb16](#) = 0x00800000, [PixelFormatRaw8](#) = 0x00400000, [PixelFormatRaw16](#) = 0x00200000, [PixelFormatMono12](#) = 0x00100000, [PixelFormatRaw12](#) = 0x00080000, [PixelFormatBgr](#) = 0x80000008, [PixelFormatBgru](#) = 0x40000008, [PixelFormatRgb](#) = [PixelFormatRgb8](#), [PixelFormatRgbu](#) = 0x40000002, [PixelFormatBgr16](#) = 0x02000001, [PixelFormatBgru16](#) = 0x02000002, [PixelFormat422Yuv8Jpeg](#) = 0x40000001, [NumberOfPixelFormats](#) = 20 }

*Pixel formats available for Format7 modes.*

- enum [BusSpeed](#) { [S100](#), [S200](#), [S400](#), [S480](#), [S800](#), [S1600](#), [S3200](#), [S5000](#), [GigE\\_10Base\\_T](#), [GigE\\_100Base\\_T](#), [GigE\\_1000Base\\_T](#), [GigE\\_10000Base\\_T](#), [Fastest](#), [Any](#), [Unknown](#) = -1 }

*Bus speeds.*

- enum [PCleBusSpeed](#) { [Speed\\_2\\_5](#), [Speed\\_5\\_0](#), [Unknown](#) = -1 }

*PCle Bus Speeds.*

- enum [DriverType](#) { [Ieee1394\\_Cam](#), [Ieee1394\\_Pro](#), [Ieee1394\\_Juju](#), [Ieee1394\\_Video1394](#), [Ieee1394\\_Raw1394](#), [Usb\\_None](#), [Usb\\_Cam](#), [Usb3\\_Pro](#), [GigE\\_None](#), [GigE\\_Filter](#), [GigE\\_Pro](#), [GigE\\_Lwf](#), [Unknown](#) = -1 }

*Types of low level drivers that flycapture uses.*

- enum [ColorProcessingAlgorithm](#) { [Default](#), [NoColorProcessing](#), [NearestNeighbor](#), [EdgeSensing](#), [HQLinear](#), [Rigorous](#), [IPP](#), [Directional](#), [Weighted-Directional](#) }

*Color processing algorithms.*

- enum [BayerTileFormat](#) { [None](#) = 0, [RGGB](#), [GRBG](#), [GBRG](#), [BGRG](#) }

*Bayer tile formats.*

- enum [ImageFileFormat](#) { [FromFileExtension](#) = -1, [Pgm](#), [Ppm](#), [Bmp](#), [Jpeg](#), [Jpeg2000](#), [Tiff](#), [Png](#), [Raw](#) }

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

- enum [GigEPropertyType](#) { [Heartbeat](#), [HeartbeatTimeout](#), [PacketSize](#), [Packet-Delay](#) }

*Possible properties that can be queried from the camera.*

- enum [StatisticsChannel](#) { [Grey](#), [Red](#), [Green](#), [Blue](#), [Hue](#), [Saturation](#), [Lightness](#), [NumberOfStatisticsChannels](#) }

*Channels that allow statistics to be calculated.*

- enum [OSType](#) { [WindowsX86](#), [WindowsX64](#), [LinuxX86](#), [LinuxX64](#), [Mac](#), [UnknownOS](#) }

*Possible operating systems.*

- enum [ByteOrder](#) { [LittleEndian](#), [BigEndian](#) }

*Possible byte orders.*

## Functions

- public delegate void [EnumCallback](#) (System::IntPtr parameter, unsigned int serialNumber)

*Bus event callback function prototype.*

- public delegate void [ImageEventCallback](#) ([ManagedImage](#)<sup>^</sup> image)

*The external callback that will be used by managed consumers.*

- protected delegate void [ImageCallbackDelegate](#) (FlyCapture2::Image \*image, void \*data)

*Internal callback that we use internally so we can create the proper external callback for users.*

- public delegate void [ManagedCameraEventCallback](#) ([ManagedEventCallback-Data](#)<sup>^</sup> data)

*The external callback that will be used by managed consumers.*

- protected delegate void [ManagedCameraEventCallbackDelegate](#) (void \*data)

*Internal callback that we use internally so we can create the proper proper external callback for users.*

- unsigned long [htonl](#) (unsigned long data)
- public delegate void [AsyncCommandCallback](#) (bool retError)

*The external callback that will be used by managed consumers.*

- protected delegate void [CommandCallbackDelegate](#) (FlyCapture2::Error retError, void \*pUserData)

*Internal callback that we use internally so we can create the proper external callback for users.*

## 7.2.1 Function Documentation

### 7.2.1.1 public delegate void FlyCapture2Managed::AsyncCommandCallback ( bool retError )

The external callback that will be used by managed consumers.

### 7.2.1.2 protected delegate void FlyCapture2Managed::CommandCallbackDelegate ( FlyCapture2::Error retError, void \* pUserData )

Internal callback that we use internally so we can create the proper external callback for users.

### 7.2.1.3 public delegate void FlyCapture2Managed::EnumCallback ( System::IntPtr parameter, unsigned int serialNumber )

Bus event callback function prototype.

Defines the syntax of the callback function that is passed into RegisterCallback() and UnregisterCallback().

### 7.2.1.4 unsigned long FlyCapture2Managed::htonl ( unsigned long data )

### 7.2.1.5 protected delegate void FlyCapture2Managed::ImageCallbackDelegate ( FlyCapture2::Image \* image, void \* data )

Internal callback that we use internally so we can create the proper external callback for users.

### 7.2.1.6 public delegate void FlyCapture2Managed::ImageEventCallback ( ManagedImage^ image )

The external callback that will be used by managed consumers.

7.2.1.7 public delegate void FlyCapture2Managed::ManagedCameraEventCallback ( ManagedEventCallbackData^ *data* )

The external callback that will be used by managed consumers.

7.2.1.8 protected delegate void FlyCapture2Managed::ManagedCameraEventCallbackDelegate ( void \* *data* )

Internal callback that we use internally so we can create the proper proper external callback for users.

## 7.3 FlyCapture2Managed::Gui Namespace Reference

### Classes

- class [CameraControlDialog](#)  
*CameraControlDialog*: managed wrapper of FlyCapture2::CameraControlDialog (see for details)
- class [CameraSelectionDialog](#)  
*CameraControlDialog*: managed wrapper of FlyCapture2::CameraSelectionDialog (see for details)





## Chapter 8

# Class Documentation

### 8.1 BMPOption Struct Reference

Options for saving Bitmap image.

#### Public Member Functions

- [BMPOption](#) ()

#### Properties

- bool [indexedColor\\_8bit](#)

#### 8.1.1 Detailed Description

Options for saving Bitmap image.

#### 8.1.2 Constructor & Destructor Documentation

8.1.2.1 [BMPOption](#) ( ) `[inline]`

#### 8.1.3 Property Documentation

8.1.3.1 bool [indexedColor\\_8bit](#)

### 8.2 CameraControlDialog Class Reference

[CameraControlDialog](#): managed wrapper of FlyCapture2::CameraControlDialog (see for details)

## Public Member Functions

- [CameraControlDialog](#) ()
- [~CameraControlDialog](#) ()
- void [Connect](#) ([ManagedCameraBase](#)^ camera)
- void [Disconnect](#) ()
- void [Show](#) ()
- void [Hide](#) ()
- bool [IsVisible](#) ()
- void [SetTitle](#) (System::String^ title)

### 8.2.1 Detailed Description

[CameraControlDialog](#): managed wrapper of FlyCapture2::CameraControlDialog (see for details)

### 8.2.2 Constructor & Destructor Documentation

#### 8.2.2.1 [CameraControlDialog](#) ( )

#### 8.2.2.2 [~CameraControlDialog](#) ( )

### 8.2.3 Member Function Documentation

#### 8.2.3.1 void [Connect](#) ( [FlyCapture2Managed::ManagedCameraBase](#)^ camera )

#### 8.2.3.2 void [Disconnect](#) ( void )

#### 8.2.3.3 void [Hide](#) ( )

#### 8.2.3.4 bool [IsVisible](#) ( )

#### 8.2.3.5 void [SetTitle](#) ( System::String^ title )

#### 8.2.3.6 void [Show](#) ( )

## 8.3 CameraInfo Struct Reference

Camera information.

## Properties

- unsigned int [serialNumber](#)  
*Device serial number.*
- [InterfaceType](#) [interfaceType](#)

*Interface type.*

- [DriverType driverType](#)  
*Driver type.*
- bool [isColorCamera](#)  
*Flag indicating if this is a color camera.*
- System::String^ [modelName](#)  
*Device model name.*
- System::String^ [vendorName](#)  
*Device vendor name.*
- System::String^ [sensorInfo](#)  
*String detailing the sensor information.*
- System::String^ [sensorResolution](#)  
*String providing the sensor resolution.*
- System::String^ [driverName](#)  
*Driver name of driver being used.*
- System::String^ [firmwareVersion](#)  
*Firmware version of camera.*
- System::String^ [firmwareBuildTime](#)  
*Firmware build time.*
- [BusSpeed maximumBusSpeed](#)  
*Maximum bus speed.*
- [PCleBusSpeed pcieBusSpeed](#)  
*Maximum PCIe bus speed.*
- [BayerTileFormat bayerTileFormat](#)  
*Bayer tile format.*
- unsigned short [busNumber](#)  
*Bus Number, set to 0 for USB and GigE.*
- unsigned short [nodeNumber](#)  
*Node Number, set to 0 for USB and GigE.*

#### IIDC specific information

- unsigned int [iidcVersion](#)  
*DCAM version.*
- [ConfigROM configROM](#)  
*Configuration ROM data.*

#### GigE specific information

- unsigned int [gigEMajorVersion](#)  
*GigE Vision version.*
- unsigned int [gigEMinorVersion](#)  
*GigE Vision minor version.*
- System::String^ [userDefinedName](#)  
*User defined name.*
- System::String^ [xmlURL1](#)

- XML URL 1.*
- System::String^ [xmlURL2](#)
- XML URL 2.*
- System::Net::NetworkInformation::PhysicalAddress^ [macAddress](#)  
*MAC address.*
- System::Net::IPAddress^ [ipAddress](#)  
*IP address.*
- System::Net::IPAddress^ [subnetMask](#)  
*Subnet mask.*
- System::Net::IPAddress^ [defaultGateway](#)  
*Default gateway.*
- unsigned int [ccpStatus](#)  
*Status/Content of CCP register.*
- unsigned int [applicationIPAddress](#)  
*Local Application IP Address.*
- unsigned int [applicationPort](#)  
*Local Application port.*

### 8.3.1 Detailed Description

Camera information.

### 8.3.2 Property Documentation

#### 8.3.2.1 unsigned int applicationIPAddress

Local Application IP Address.

#### 8.3.2.2 unsigned int applicationPort

Local Application port.

#### 8.3.2.3 BayerTileFormat bayerTileFormat

Bayer tile format.

#### 8.3.2.4 unsigned short busNumber

Bus Number, set to 0 for USB and GigE.

#### 8.3.2.5 unsigned int ccpStatus

Status/Content of CCP register.

**8.3.2.6 ConfigROM configROM**

Configuration ROM data.

**8.3.2.7 System:: Net:: IPAddress^ defaultGateway**

Default gateway.

**8.3.2.8 System:: String^ driverName**

Driver name of driver being used.

**8.3.2.9 DriverType driverType**

Driver type.

**8.3.2.10 System:: String^ firmwareBuildTime**

Firmware build time.

**8.3.2.11 System:: String^ firmwareVersion**

Firmware version of camera.

**8.3.2.12 unsigned int gigEMajorVersion**

GigE Vision version.

**8.3.2.13 unsigned int gigEMinorVersion**

GigE Vision minor version.

**8.3.2.14 unsigned int iidcVersion**

DCAM version.

**8.3.2.15 InterfaceType interfaceType**

Interface type.

**8.3.2.16 System:: Net:: IPAddress^ ipAddress**

IP address.

**8.3.2.17 bool isColorCamera**

Flag indicating if this is a color camera.

**8.3.2.18 System:: Net:: NetworkInformation:: PhysicalAddress^ macAddress**

MAC address.

**8.3.2.19 BusSpeed maximumBusSpeed**

Maximum bus speed.

**8.3.2.20 System:: String^ modelName**

Device model name.

**8.3.2.21 unsigned short nodeNumber**

Node Number, set to 0 for USB and GigE.

**8.3.2.22 PCleBusSpeed pcieBusSpeed**

Maximum PCIe bus speed.

**8.3.2.23 System:: String^ sensorInfo**

String detailing the sensor information.

**8.3.2.24 System:: String^ sensorResolution**

String providing the sensor resolution.

**8.3.2.25 unsigned int serialNumber**

Device serial number.

8.3.2.26 System:: Net:: IPAddress^ subnetMask

Subnet mask.

8.3.2.27 System:: String^ userDefinedName

User defined name.

8.3.2.28 System:: String^ vendorName

Device vendor name.

8.3.2.29 System:: String^ xmlURL1

XML URL 1.

8.3.2.30 System:: String^ xmlURL2

XML URL 2.

## 8.4 CameraProperty Struct Reference

A specific camera property.

### Public Member Functions

- [CameraProperty](#) ()
- [CameraProperty](#) (PropertyType type)

### Properties

- [PropertyType](#) type  
*Property info type.*
- bool [present](#)  
*Flag indicating if the property is present.*
- bool [absControl](#)  
*Flag controlling absolute mode.*
- bool [onePush](#)  
*Flag controlling one push.*
- bool [onOff](#)  
*Flag controlling on/off.*

- bool `autoManualMode`  
*Flag controlling auto.*
- unsigned int `valueA`  
*Value A (integer).*
- unsigned int `valueB`  
*Value B (integer).*
- float `absValue`  
*Floating point value.*

### 8.4.1 Detailed Description

A specific camera property.

### 8.4.2 Constructor & Destructor Documentation

8.4.2.1 `CameraProperty ( )` `[inline]`

8.4.2.2 `CameraProperty ( PropertyType type )` `[inline]`

### 8.4.3 Property Documentation

8.4.3.1 bool `absControl`

Flag controlling absolute mode.

8.4.3.2 float `absValue`

Floating point value.

8.4.3.3 bool `autoManualMode`

Flag controlling auto.

8.4.3.4 bool `onePush`

Flag controlling one push.

8.4.3.5 bool `onOff`

Flag controlling on/off.



#### 8.4.3.6 bool present

Flag indicating if the property is present.

#### 8.4.3.7 PropertyType type

Property info type.

#### 8.4.3.8 unsigned int valueA

Value A (integer).

#### 8.4.3.9 unsigned int valueB

Value B (integer).

Applies only to the white balance blue value. Use Value A for the red value.

## 8.5 CameraPropertyInfo Struct Reference

Information about a specific camera property.

### Public Member Functions

- [CameraPropertyInfo \(\)](#)
- [CameraPropertyInfo \(PropertyType type\)](#)

### Properties

- [PropertyType type](#)  
*Property info type.*
- bool [present](#)  
*Flag indicating if the property is present.*
- bool [autoSupported](#)  
*Flag indicating if auto is supported.*
- bool [manualSupported](#)  
*Flag indicating if manual is supported.*
- bool [onOffSupported](#)  
*Flag indicating if on/off is supported.*
- bool [onePushSupported](#)  
*Flag indicating if one push is supported.*
- bool [absValSupported](#)

*Flag indicating if absolute mode is supported.*

- bool `readOutSupported`

*Flag indicating if property value can be read out.*

- unsigned int `min`

*Minimum value (as an integer).*

- unsigned int `max`

*Maximum value (as an integer).*

- float `absMin`

*Minimum value (as a floating point value).*

- float `absMax`

*Maximum value (as a floating point value).*

- System::String^ `units`

*Textual description of units.*

- System::String^ `unitAbbr`

*Abbreviated textual description of units.*

### 8.5.1 Detailed Description

Information about a specific camera property.

This structure is also also used as the TriggerDelayInfo structure.

### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 CameraPropertyInfo ( ) [inline]

#### 8.5.2.2 CameraPropertyInfo ( PropertyType type ) [inline]

### 8.5.3 Property Documentation

#### 8.5.3.1 float absMax

Maximum value (as a floating point value).

#### 8.5.3.2 float absMin

Minimum value (as a floating point value).

#### 8.5.3.3 bool absValSupported

Flag indicating if absolute mode is supported.

**8.5.3.4 bool autoSupported**

Flag indicating if auto is supported.

**8.5.3.5 bool manualSupported**

Flag indicating if manual is supported.

**8.5.3.6 unsigned int max**

Maximum value (as an integer).

**8.5.3.7 unsigned int min**

Minimum value (as an integer).

**8.5.3.8 bool onePushSupported**

Flag indicating if one push is supported.

**8.5.3.9 bool onOffSupported**

Flag indicating if on/off is supported.

**8.5.3.10 bool present**

Flag indicating if the property is present.

**8.5.3.11 bool readOutSupported**

Flag indicating if property value can be read out.

**8.5.3.12 PropertyType type**

Property info type.

**8.5.3.13 System::String^ unitAbbr**

Abbreviated textual description of units.

#### 8.5.3.14 System::String^ units

Textual description of units.

## 8.6 CameraSelectionDialog Class Reference

[CameraControlDialog](#): managed wrapper of FlyCapture2::CameraSelectionDialog (see for details)

### Public Member Functions

- [CameraSelectionDialog](#) ()
- [~CameraSelectionDialog](#) ()
- bool [ShowModal](#) ()  
*Show the CameraSelectionDlg.*
- array< [ManagedPGRGuid](#)^ >^ [GetSelectedCameraGuids](#) ()  
*Returns the list of camera guides selected by the user while in [ShowModal\(\)](#)*
- void [SetTitle](#) (System::String^ title)  
*Set the window title.*

#### 8.6.1 Detailed Description

[CameraControlDialog](#): managed wrapper of FlyCapture2::CameraSelectionDialog (see for details)

#### 8.6.2 Constructor & Destructor Documentation

##### 8.6.2.1 CameraSelectionDialog ( )

##### 8.6.2.2 ~CameraSelectionDialog ( )

#### 8.6.3 Member Function Documentation

##### 8.6.3.1 array< ManagedPGRGuid^ > GetSelectedCameraGuids ( )

Returns the list of camera guides selected by the user while in [ShowModal\(\)](#)

#### Returns

Array of PGRGuids identifying the selected cameras.

### 8.6.3.2 void SetTitle ( System::String^ title )

Set the window title.

#### Parameters

<i>pTitle</i>	Null-terminated string representing the title.
---------------	--

### 8.6.3.3 bool ShowModal ( )

Show the CameraSelectionDlg.

#### Returns

Whether Ok (true) or Cancel (false) was clicked.

## 8.7 CameraStats Struct Reference

Camera diagnostic information.

### Public Member Functions

- [CameraStats](#) ()

### Properties

- unsigned int [imageDropped](#)
- unsigned int [imageCorrupt](#)
- unsigned int [imageXmitFailed](#)
- unsigned int [imageDriverDropped](#)
- unsigned int [regReadFailed](#)
- unsigned int [regWriteFailed](#)
- unsigned int [portErrors](#)
- bool [cameraPowerUp](#)
- List< float >^ [cameraVoltages](#)
- unsigned int [numVoltages](#)  
*The number of voltage registers available.*
- List< float >^ [cameraCurrents](#)
- unsigned int [numCurrents](#)  
*The number of current registers available.*
- unsigned int [temperature](#)
- unsigned int [timeSinceInitialization](#)
- unsigned int [timeSinceBusReset](#)
- [TimeStamp](#)^ [timeStamp](#)
- unsigned int [numResendPacketsRequested](#)
- unsigned int [numResendPacketsReceived](#)

### 8.7.1 Detailed Description

Camera diagnostic information.

### 8.7.2 Constructor & Destructor Documentation

8.7.2.1 **CameraStats**( ) [inline]

### 8.7.3 Property Documentation

8.7.3.1 **List< float>^ cameraCurrents**

8.7.3.2 **bool cameraPowerUp**

8.7.3.3 **List< float>^ cameraVoltages**

8.7.3.4 **unsigned int imageCorrupt**

8.7.3.5 **unsigned int imageDriverDropped**

8.7.3.6 **unsigned int imageDropped**

8.7.3.7 **unsigned int imageXmitFailed**

8.7.3.8 **unsigned int numCurrents**

The number of current registers available.

0: the values in cameraCurrents[] are invalid.

8.7.3.9 **unsigned int numResendPacketsReceived**

8.7.3.10 **unsigned int numResendPacketsRequested**

8.7.3.11 **unsigned int numVoltages**

The number of voltage registers available.

0: the values in cameraVoltages[] are invalid.

8.7.3.12 **unsigned int portErrors**

8.7.3.13 **unsigned int regReadFailed**

8.7.3.14 **unsigned int regWriteFailed**

8.7.3.15 unsigned int temperature

8.7.3.16 unsigned int timeSinceBusReset

8.7.3.17 unsigned int timeSinceInitialization

8.7.3.18 TimeStamp<sup>^</sup> timeStamp

## 8.8 ConfigROM Struct Reference

Camera configuration ROM.

### Properties

- unsigned int [nodeVendorId](#)  
*Vendor ID of a node.*
- unsigned int [chipIdHi](#)  
*Chip ID (high part).*
- unsigned int [chipIdLo](#)  
*Chip ID (low part).*
- unsigned int [unitSpecId](#)  
*Unit Spec ID, usually 0xa02d.*
- unsigned int [unitSWVer](#)  
*Unit software version.*
- unsigned int [unitSubSWVer](#)  
*Unit sub software version.*
- unsigned int [vendorUniqueInfo0](#)  
*Vendor unique info 0.*
- unsigned int [vendorUniqueInfo1](#)  
*Vendor unique info 1.*
- unsigned int [vendorUniqueInfo2](#)  
*Vendor unique info 2.*
- unsigned int [vendorUniqueInfo3](#)  
*Vendor unique info 3.*
- System::String<sup>^</sup> [keyword](#)  
*Keyword.*

### 8.8.1 Detailed Description

Camera configuration ROM.

## 8.8.2 Property Documentation

### 8.8.2.1 unsigned int chipIdHi

Chip ID (high part).

### 8.8.2.2 unsigned int chipIdLo

Chip ID (low part).

### 8.8.2.3 System::String<sup>^</sup> keyword

Keyword.

### 8.8.2.4 unsigned int nodeVendorId

Vendor ID of a node.

### 8.8.2.5 unsigned int unitSpecId

Unit Spec ID, usually 0xa02d.

### 8.8.2.6 unsigned int unitSubSWVer

Unit sub software version.

### 8.8.2.7 unsigned int unitSWVer

Unit software version.

### 8.8.2.8 unsigned int vendorUniqueInfo0

Vendor unique info 0.

### 8.8.2.9 unsigned int vendorUniqueInfo1

Vendor unique info 1.

### 8.8.2.10 unsigned int vendorUniqueInfo2

Vendor unique info 2.



### 8.8.2.11 unsigned int vendorUniqueInfo3

Vendor unique info 3.

## 8.9 EmbeddedImageInfo Struct Reference

Properties of the possible embedded image information.

### Public Member Functions

- [EmbeddedImageInfo](#) ()

### Properties

- [EmbeddedImageInfoProperty](#)^ timestamp
- [EmbeddedImageInfoProperty](#)^ gain
- [EmbeddedImageInfoProperty](#)^ shutter
- [EmbeddedImageInfoProperty](#)^ brightness
- [EmbeddedImageInfoProperty](#)^ exposure
- [EmbeddedImageInfoProperty](#)^ whiteBalance
- [EmbeddedImageInfoProperty](#)^ frameCounter
- [EmbeddedImageInfoProperty](#)^ strobePattern
- [EmbeddedImageInfoProperty](#)^ GPIOPinState
- [EmbeddedImageInfoProperty](#)^ ROIPosition

### 8.9.1 Detailed Description

Properties of the possible embedded image information.

### 8.9.2 Constructor & Destructor Documentation

#### 8.9.2.1 [EmbeddedImageInfo](#) ( ) [inline]

### 8.9.3 Property Documentation

#### 8.9.3.1 [EmbeddedImageInfoProperty](#)^ brightness

#### 8.9.3.2 [EmbeddedImageInfoProperty](#)^ exposure

#### 8.9.3.3 [EmbeddedImageInfoProperty](#)^ frameCounter

#### 8.9.3.4 [EmbeddedImageInfoProperty](#)^ gain

8.9.3.5 `EmbeddedImageInfoProperty`^ `GPIOPinState`

8.9.3.6 `EmbeddedImageInfoProperty`^ `ROIPosition`

8.9.3.7 `EmbeddedImageInfoProperty`^ `shutter`

8.9.3.8 `EmbeddedImageInfoProperty`^ `strobePattern`

8.9.3.9 `EmbeddedImageInfoProperty`^ `timestamp`

8.9.3.10 `EmbeddedImageInfoProperty`^ `whiteBalance`

## 8.10 `EmbeddedImageInfoProperty` Struct Reference

Properties of a single embedded image info property.

### Properties

- bool `available`  
*Whether this property is available.*
- bool `onOff`  
*Whether this property is on or off.*

### 8.10.1 Detailed Description

Properties of a single embedded image info property.

### 8.10.2 Property Documentation

#### 8.10.2.1 bool `available`

Whether this property is available.

#### 8.10.2.2 bool `onOff`

Whether this property is on or off.

## 8.11 `FC2Config` Struct Reference

Configuration for a camera.

## Public Member Functions

- [FC2Config](#) ()

## Properties

- unsigned int [numBuffers](#)  
*Number of buffers used by the [FlyCapture2](#) library to grab images.*
- unsigned int [numImageNotifications](#)  
*Number of notifications per image.*
- unsigned int [minNumImageNotifications](#)  
*Minimum number of notifications needed for the current image settings on the camera.*
- int [grabTimeout](#)  
*Time in milliseconds that [RetrieveBuffer\(\)](#) and [WaitForBufferEvent\(\)](#) will wait for an image before timing out and returning.*
- [GrabMode](#) [grabMode](#)  
*Grab mode for the camera.*
- bool [highPerformanceRetrieveBuffer](#)  
*This parameter enables [RetrieveBuffer](#) to run in high performance mode.*
- [BusSpeed](#) [isochBusSpeed](#)  
*Isochronous bus speed.*
- [BusSpeed](#) [asyncBusSpeed](#)  
*Asynchronous bus speed.*
- [BandwidthAllocation](#) [bandwidthAllocation](#)  
*Bandwidth allocation flag that tells the camera the bandwidth allocation strategy to employ.*
- unsigned int [registerTimeoutRetries](#)  
*Number of retries to perform when a register read/write timeout is received by the library.*
- unsigned int [registerTimeout](#)  
*Register read/write timeout value, in microseconds.*

### 8.11.1 Detailed Description

Configuration for a camera.

These options are options that are generally should be set before starting isochronous transfer.

### 8.11.2 Constructor & Destructor Documentation

#### 8.11.2.1 [FC2Config](#) ( ) [\[inline\]](#)

### 8.11.3 Property Documentation

#### 8.11.3.1 **BusSpeed** `asyncBusSpeed`

Asynchronous bus speed.

#### 8.11.3.2 **BandwidthAllocation** `bandwidthAllocation`

Bandwidth allocation flag that tells the camera the bandwidth allocation strategy to employ.

#### 8.11.3.3 **GrabMode** `grabMode`

Grab mode for the camera.

The default is DROP\_FRAMES.

#### 8.11.3.4 **int** `grabTimeout`

Time in milliseconds that `RetrieveBuffer()` and `WaitForBufferEvent()` will wait for an image before timing out and returning.

#### 8.11.3.5 **bool** `highPerformanceRetrieveBuffer`

This parameter enables `RetrieveBuffer` to run in high performance mode.

This means that any interaction with the camera, other than grabbing the image is disabled. Currently `RetrieveBuffer` reads registers on the camera to determine which embedded image information settings have been enabled, and it reads what the bayer tile is currently set to. When High Performance mode is on, these reads are disabled. - This means that any changes to the Bayer Tile or to the Embedded image info after `StartCapture()` will not be tracked when made using direct register writes. If the corresponding `SetEmbeddedImageInfo()` and `GetEmbeddedImageInfo()` calls are used then the changes will be appropriately reflected. This also means that changes to embedded image info from other processes will not be updated either.

#### 8.11.3.6 **BusSpeed** `isochBusSpeed`

Isochronous bus speed.

#### 8.11.3.7 **unsigned int** `minNumImageNotifications`

Minimum number of notifications needed for the current image settings on the camera.

Read-only value.

### 8.11.3.8 unsigned int numBuffers

Number of buffers used by the [FlyCapture2](#) library to grab images.

### 8.11.3.9 unsigned int numImageNotifications

Number of notifications per image.

This value should only be set after the image settings to be used is set to the camera.  
The default number of notifications is 1.

There are 4 general scenarios:

- 1 notification - End of image
- 2 notifications - After first packet and end of image
- 3 notifications - After first packet, middle of image, end of image
- x notifications - After first packet, (x -2) spread evenly, end of image

### 8.11.3.10 unsigned int registerTimeout

Register read/write timeout value, in microseconds.

The default value is dependent on the interface type.

### 8.11.3.11 unsigned int registerTimeoutRetries

Number of retries to perform when a register read/write timeout is received by the library.

The default value is 0.

## 8.12 FC2Exception Class Reference

Exception that is thrown when an error is encountered.

### Public Member Functions

- [FC2Exception](#) ()
- [FC2Exception](#) (String^ string)
- [FC2Exception](#) (String^ string, Exception^ exception)
- [~FC2Exception](#) ()

## Protected Member Functions

- [FC2Exception](#) (Runtime::Serialization::SerializationInfo^ serializationInfo, - Runtime::Serialization::StreamingContext context)

## Package Functions

- [FC2Exception](#) (FlyCapture2::Error \*error)

## Properties

- [ErrorType Type](#) [get]
- [ErrorType CauseType](#) [get]
- String^ [NativeErrorTrace](#) [get]

### 8.12.1 Detailed Description

Exception that is thrown when an error is encountered.

This is used instead of returning an Error object as used in the C++ interface.

### 8.12.2 Constructor & Destructor Documentation

8.12.2.1 [FC2Exception](#) ( )

8.12.2.2 [FC2Exception](#) ( String^ *string* )

8.12.2.3 [FC2Exception](#) ( String^ *string*, Exception^ *exception* )

8.12.2.4 [~FC2Exception](#) ( )

8.12.2.5 [FC2Exception](#) ( Runtime::Serialization::SerializationInfo^ *serializationInfo*, Runtime::Serialization::StreamingContext *context* ) [protected]

8.12.2.6 [FC2Exception](#) ( FlyCapture2::Error \* *error* ) [package]

### 8.12.3 Property Documentation

8.12.3.1 [ErrorType CauseType](#) [get]

8.12.3.2 [String^ NativeErrorTrace](#) [get]

8.12.3.3 [ErrorType Type](#) [get]

## 8.13 FC2Version Struct Reference

The current version of the library.

### Properties

- unsigned int [major](#)  
*Major version number.*
- unsigned int [minor](#)  
*Minor version number.*
- unsigned int [type](#)  
*Type version number.*
- unsigned int [build](#)  
*Build version number.*

### 8.13.1 Detailed Description

The current version of the library.

### 8.13.2 Property Documentation

#### 8.13.2.1 unsigned int build

Build version number.

#### 8.13.2.2 unsigned int major

Major version number.

#### 8.13.2.3 unsigned int minor

Minor version number.

#### 8.13.2.4 unsigned int type

Type version number.

## 8.14 Format7ImageSettings Struct Reference

Format 7 image settings.

## Properties

- [Mode mode](#)  
*Format 7 mode.*
- unsigned int [offsetX](#)  
*Horizontal image offset.*
- unsigned int [offsetY](#)  
*Vertical image offset.*
- unsigned int [width](#)  
*Width of image.*
- unsigned int [height](#)  
*Height of image.*
- [PixelFormat pixelFormat](#)  
*Pixel format of image.*

### 8.14.1 Detailed Description

Format 7 image settings.

### 8.14.2 Property Documentation

#### 8.14.2.1 unsigned int height

Height of image.

#### 8.14.2.2 Mode mode

Format 7 mode.

#### 8.14.2.3 unsigned int offsetX

Horizontal image offset.

#### 8.14.2.4 unsigned int offsetY

Vertical image offset.

#### 8.14.2.5 PixelFormat pixelFormat

Pixel format of image.



## 8.14.2.6 unsigned int width

Width of image.

## 8.15 Format7Info Struct Reference

Format 7 information for a single mode.

### Properties

- [Mode mode](#)  
*Format 7 mode.*
- unsigned int [maxWidth](#)  
*Maximum image width.*
- unsigned int [maxHeight](#)  
*Maximum image height.*
- unsigned int [offsetHStepSize](#)  
*Horizontal step size for the offset.*
- unsigned int [offsetVStepSize](#)  
*Vertical step size for the offset.*
- unsigned int [imageHStepSize](#)  
*Horizontal step size for the image.*
- unsigned int [imageVStepSize](#)  
*Vertical step size for the image.*
- unsigned int [pixelFormatBitField](#)  
*Supported pixel formats in a bit field.*
- unsigned int [vendorPixelFormatBitField](#)  
*Vendor unique pixel formats in a bit field.*
- unsigned int [packetSize](#)  
*Current packet size in bytes.*
- unsigned int [minPacketSize](#)  
*Minimum packet size in bytes for current mode.*
- unsigned int [maxPacketSize](#)  
*Maximum packet size in bytes for current mode.*
- float [percentage](#)  
*Current packet size as a percentage of maximum packet size.*

### 8.15.1 Detailed Description

Format 7 information for a single mode.

## 8.15.2 Property Documentation

### 8.15.2.1 unsigned int imageHStepSize

Horizontal step size for the image.

### 8.15.2.2 unsigned int imageVStepSize

Vertical step size for the image.

### 8.15.2.3 unsigned int maxHeight

Maximum image height.

### 8.15.2.4 unsigned int maxPacketSize

Maximum packet size in bytes for current mode.

### 8.15.2.5 unsigned int maxWidth

Maximum image width.

### 8.15.2.6 unsigned int minPacketSize

Minimum packet size in bytes for current mode.

### 8.15.2.7 **Mode** mode

Format 7 mode.

### 8.15.2.8 unsigned int offsetHStepSize

Horizontal step size for the offset.

### 8.15.2.9 unsigned int offsetVStepSize

Vertical step size for the offset.

### 8.15.2.10 unsigned int packetSize

Current packet size in bytes.

## 8.15.2.11 float percentage

Current packet size as a percentage of maximum packet size.

## 8.15.2.12 unsigned int pixelFormatBitField

Supported pixel formats in a bit field.

## 8.15.2.13 unsigned int vendorPixelFormatBitField

Vendor unique pixel formats in a bit field.

## 8.16 Format7PacketInfo Struct Reference

Format 7 packet information.

### Properties

- unsigned int [recommendedBytesPerPacket](#)  
*Recommended bytes per packet.*
- unsigned int [maxBytesPerPacket](#)  
*Maximum bytes per packet.*
- unsigned int [unitBytesPerPacket](#)  
*Minimum bytes per packet.*

### 8.16.1 Detailed Description

Format 7 packet information.

### 8.16.2 Property Documentation

## 8.16.2.1 unsigned int maxBytesPerPacket

Maximum bytes per packet.

## 8.16.2.2 unsigned int recommendedBytesPerPacket

Recommended bytes per packet.

## 8.16.2.3 unsigned int unitBytesPerPacket

Minimum bytes per packet.

## 8.17 GigEConfig Struct Reference

Configuration for a GigE camera.

### Properties

- bool [enablePacketResend](#)  
*Turn on/off packet resend functionality.*

### 8.17.1 Detailed Description

Configuration for a GigE camera.

These options are options that are generally should be set before starting isochronous transfer.

### 8.17.2 Property Documentation

#### 8.17.2.1 bool [enablePacketResend](#)

Turn on/off packet resend functionality.

## 8.18 GigEImageSettings Struct Reference

Image settings for a GigE camera.

### Properties

- unsigned int [offsetX](#)  
*Horizontal image offset.*
- unsigned int [offsetY](#)  
*Vertical image offset.*
- unsigned int [width](#)  
*Width of image.*
- unsigned int [height](#)  
*Height of image.*
- [PixelFormat](#) [pixelFormat](#)  
*Pixel format of image.*

### 8.18.1 Detailed Description

Image settings for a GigE camera.

### 8.18.2 Property Documentation

#### 8.18.2.1 unsigned int height

Height of image.

#### 8.18.2.2 unsigned int offsetX

Horizontal image offset.

#### 8.18.2.3 unsigned int offsetY

Vertical image offset.

#### 8.18.2.4 PixelFormat pixelFormat

Pixel format of image.

#### 8.18.2.5 unsigned int width

Width of image.

## 8.19 GigImageSettingsInfo Struct Reference

Format 7 information for a single mode.

### Properties

- unsigned int [maxWidth](#)  
*Maximum image width.*
- unsigned int [maxHeight](#)  
*Maximum image height.*
- unsigned int [offsetHStepSize](#)  
*Horizontal step size for the offset.*
- unsigned int [offsetVStepSize](#)  
*Vertical step size for the offset.*
- unsigned int [imageHStepSize](#)  
*Horizontal step size for the image.*
- unsigned int [imageVStepSize](#)  
*Vertical step size for the image.*
- unsigned int [pixelFormatBitField](#)  
*Supported pixel formats in a bit field.*

- unsigned int [vendorPixelFormatBitField](#)  
*Vendor unique pixel formats in a bit field.*

### 8.19.1 Detailed Description

Format 7 information for a single mode.

### 8.19.2 Property Documentation

#### 8.19.2.1 unsigned int imageHStepSize

Horizontal step size for the image.

#### 8.19.2.2 unsigned int imageVStepSize

Vertical step size for the image.

#### 8.19.2.3 unsigned int maxHeight

Maximum image height.

#### 8.19.2.4 unsigned int maxWidth

Maximum image width.

#### 8.19.2.5 unsigned int offsetHStepSize

Horizontal step size for the offset.

#### 8.19.2.6 unsigned int offsetVStepSize

Vertical step size for the offset.

#### 8.19.2.7 unsigned int pixelFormatBitField

Supported pixel formats in a bit field.

#### 8.19.2.8 unsigned int vendorPixelFormatBitField

Vendor unique pixel formats in a bit field.

## 8.20 GigEProperty Struct Reference

A GigE property.

### Properties

- [GigEPropertyType propType](#)  
*The type of property.*
- bool [isReadable](#)  
*Whether the property is readable.*
- bool [isWritable](#)  
*Whether the property is writable.*
- unsigned int [min](#)  
*Minimum value.*
- unsigned int [max](#)  
*Maximum value.*
- unsigned int [value](#)  
*Current value.*

### 8.20.1 Detailed Description

A GigE property.

### 8.20.2 Property Documentation

#### 8.20.2.1 bool isReadable

Whether the property is readable.

If this is false, then no other value in this structure is valid.

#### 8.20.2.2 bool isWritable

Whether the property is writable.

#### 8.20.2.3 unsigned int max

Maximum value.

#### 8.20.2.4 unsigned int min

Minimum value.

#### 8.20.2.5 GigEPropertyType propType

The type of property.

#### 8.20.2.6 unsigned int value

Current value.

## 8.21 GigEStreamChannel Struct Reference

Information about a single GigE stream channel.

### Properties

- unsigned int [networkInterfaceIndex](#)  
*Network interface index used (or to use).*
- unsigned int [hostPort](#)  
*Host port on the PC where the camera will send the data stream.*
- bool [doNotFragment](#)  
*Disable IP fragmentation of packets.*
- unsigned int [packetSize](#)  
*Packet size, in bytes.*
- unsigned int [interPacketDelay](#)  
*Inter packet delay, in timestamp counter units.*
- System::Net::IPAddress^ [destinationIpAddress](#)  
*Destination IP address.*
- unsigned int [sourcePort](#)  
*Source UDP port of the stream channel.*

### 8.21.1 Detailed Description

Information about a single GigE stream channel.

### 8.21.2 Property Documentation

#### 8.21.2.1 System::Net::IPAddress^ destinationIpAddress

Destination IP address.

It can be a multicast or unicast address.



### 8.21.2.2 bool doNotFragment

Disable IP fragmentation of packets.

### 8.21.2.3 unsigned int hostPort

Host port on the PC where the camera will send the data stream.

### 8.21.2.4 unsigned int interPacketDelay

Inter packet delay, in timestamp counter units.

### 8.21.2.5 unsigned int networkInterfaceIndex

Network interface index used (or to use).

### 8.21.2.6 unsigned int packetSize

Packet size, in bytes.

### 8.21.2.7 unsigned int sourcePort

Source UDP port of the stream channel.

Read only.

## 8.22 ImageMetadata Struct Reference

Metadata related to an image.

### Properties

- unsigned int [embeddedTimeStamp](#)  
*Embedded timestamp.*
- unsigned int [embeddedGain](#)  
*Embedded gain.*
- unsigned int [embeddedShutter](#)  
*Embedded shutter.*
- unsigned int [embeddedBrightness](#)  
*Embedded brightness.*
- unsigned int [embeddedExposure](#)  
*Embedded exposure.*

- unsigned int [embeddedWhiteBalance](#)  
*Embedded white balance.*
- unsigned int [embeddedFrameCounter](#)  
*Embedded frame counter.*
- unsigned int [embeddedStrobePattern](#)  
*Embedded strobe pattern.*
- unsigned int [embeddedGPIOPinState](#)  
*Embedded GPIO pin state.*
- unsigned int [embeddedROIPosition](#)  
*Embedded ROI position.*

### 8.22.1 Detailed Description

Metadata related to an image.

### 8.22.2 Property Documentation

#### 8.22.2.1 unsigned int embeddedBrightness

Embedded brightness.

#### 8.22.2.2 unsigned int embeddedExposure

Embedded exposure.

#### 8.22.2.3 unsigned int embeddedFrameCounter

Embedded frame counter.

#### 8.22.2.4 unsigned int embeddedGain

Embedded gain.

#### 8.22.2.5 unsigned int embeddedGPIOPinState

Embedded GPIO pin state.

#### 8.22.2.6 unsigned int embeddedROIPosition

Embedded ROI position.

## 8.22.2.7 unsigned int embeddedShutter

Embedded shutter.

## 8.22.2.8 unsigned int embeddedStrobePattern

Embedded strobe pattern.

## 8.22.2.9 unsigned int embeddedTimeStamp

Embedded timestamp.

## 8.22.2.10 unsigned int embeddedWhiteBalance

Embedded white balance.

## 8.23 JpegOption Struct Reference

Options for saving JPEG image.

### Public Member Functions

- [JpegOption](#) ()

### Properties

- bool [progressive](#)  
*Whether to save as a progressive JPEG file.*
- unsigned int [quality](#)  
*JPEG image quality in range (0-100).*

#### 8.23.1 Detailed Description

Options for saving JPEG image.

#### 8.23.2 Constructor & Destructor Documentation

8.23.2.1 [JpegOption](#) ( ) `[inline]`

#### 8.23.3 Property Documentation

### 8.23.3.1 bool progressive

Whether to save as a progressive JPEG file.

### 8.23.3.2 unsigned int quality

JPEG image quality in range (0-100).

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

## 8.24 Jpg2Option Struct Reference

Options for saving JPEG2000 image.

### Public Member Functions

- [Jpg2Option](#) ()

### Properties

- unsigned int [quality](#)  
*JPEG saving quality in range (1-512).*

### 8.24.1 Detailed Description

Options for saving JPEG2000 image.

### 8.24.2 Constructor & Destructor Documentation

#### 8.24.2.1 Jpg2Option ( ) `[inline]`

### 8.24.3 Property Documentation

#### 8.24.3.1 unsigned int quality

JPEG saving quality in range (1-512).

## 8.25 LutData Struct Reference

Information about the camera's look up table.

### Properties

- bool [supported](#)  
*Flag indicating if LUT is supported.*
- bool [enabled](#)  
*Flag indicating if LUT is enabled.*
- unsigned int [numBanks](#)  
*The number of LUT banks available (Always 1 for PGR LUT).*
- unsigned int [numChannels](#)  
*The number of LUT channels per bank available.*
- unsigned int [inputBitDepth](#)  
*The input bit depth of the LUT.*
- unsigned int [outputBitDepth](#)  
*The output bit depth of the LUT.*
- unsigned int [numEntries](#)  
*The number of entries in the LUT.*

### 8.25.1 Detailed Description

Information about the camera's look up table.

### 8.25.2 Property Documentation

#### 8.25.2.1 bool enabled

Flag indicating if LUT is enabled.

#### 8.25.2.2 unsigned int inputBitDepth

The input bit depth of the LUT.

#### 8.25.2.3 unsigned int numBanks

The number of LUT banks available (Always 1 for PGR LUT).

#### 8.25.2.4 unsigned int numChannels

The number of LUT channels per bank available.

#### 8.25.2.5 unsigned int numEntries

The number of entries in the LUT.

#### 8.25.2.6 unsigned int outputBitDepth

The output bit depth of the LUT.

#### 8.25.2.7 bool supported

Flag indicating if LUT is supported.

## 8.26 ManagedBusManager Class Reference

[ManagedBusManager](#) provides the functionality for the user to get an [PGRGuid](#) for a desired camera or device easily.

### Public Member Functions

- [ManagedBusManager](#) ()
- [~ManagedBusManager](#) ()
- void [FireBusReset](#) ([ManagedPGRGuid](#)^ guid)  
*Fire a bus reset.*
- unsigned int [GetNumOfCameras](#) ()  
*Gets the number of cameras attached to the PC.*
- [ManagedPGRGuid](#)^ [GetCameraFromIPAddress](#) (System::Net::IPAddress^ ipAddress)  
*Gets the [ManagedPGRGuid](#) for a camera with the specified IPv4 address.*
- [ManagedPGRGuid](#)^ [GetCameraFromIndex](#) (unsigned int index)  
*Gets the [ManagedPGRGuid](#) for a camera on the PC.*
- [ManagedPGRGuid](#)^ [GetCameraFromSerialNumber](#) (unsigned int serialNumber)  
*Gets the [ManagedPGRGuid](#) for a camera on the PC.*
- unsigned int [GetCameraSerialNumberFromIndex](#) (unsigned int index)  
*Gets the serial number of the camera with the specified index.*
- [InterfaceType](#) [GetInterfaceTypeFromGuid](#) ([ManagedPGRGuid](#)^ guid)  
*Gets the interface type associated with a [ManagedPGRGuid](#).*
- unsigned int [GetNumOfDevices](#) ()  
*Gets the number of devices.*
- [ManagedPGRGuid](#)^ [GetDeviceFromIndex](#) (unsigned int index)  
*Gets the [ManagedPGRGuid](#) for a device.*
- unsigned int [ReadPhyRegister](#) ([ManagedPGRGuid](#)^ guid, unsigned int page, unsigned int port, unsigned int address)

*Read a phy register on the specified device.*

- void [WritePhyRegister](#) ([ManagedPGRGuid](#)^ guid, unsigned int page, unsigned int port, unsigned int address, unsigned int regVal)

*Write a phy register on the specified device.*

- unsigned int [GetUsbLinkInfo](#) ([ManagedPGRGuid](#)^ guid)

*Read usb link info for the port that the specified device is connected to.*

- unsigned int [GetUsbPortStatus](#) ([ManagedPGRGuid](#)^ guid)

*Read usb port status for the port that the specified device is connected to.*

- [ManagedTopologyNode](#)^ [GetTopology](#) ()

*Gets the topology information for the PC.*

- System::IntPtr [RegisterCallback](#) (EnumCallback^ hCallbackDelegate, [Managed-CallbackType](#) callbackType, System::IntPtr parameter)

*Register a callback function that will be called when the specified callback event occurs.*

- void [UnregisterCallback](#) (System::IntPtr callbackHandle)

*Unregister a callback function.*

- void [RescanBus](#) ()

*Force a rescan of the buses.*

- bool [IsCameraControlable](#) ([ManagedPGRGuid](#)^ guid)

*Query CCP status on camera with corresponding PGRGuid.*

## Static Public Member Functions

- static void [ForceIPAddressToCamera](#) (System::Net::NetworkInformation::PhysicalAddress^ macAddress, System::Net::IPAddress^ ipAddress, System::Net::IPAddress^ subnetMask, System::Net::IPAddress^ defaultGateway)

*Force the camera with the specific MAC address to the specified IP address, subnet mask and default gateway.*

- static void [ForceAllIPAddressesAutomatically](#) ()

*Force all cameras on the network to be assigned sequential IP addresses on the same subnet as the network adapters that they are connected to.*

- static void [ForceAllIPAddressesAutomatically](#) (unsigned int serialNumber)

*Force a specific cameras on the network to be assigned sequential IP address on the same subnet as the network adapters that it is connected to.*

- static array< [CameraInfo](#)^ >^ [DiscoverGigECameras](#) ()

*Discover all cameras connected to the network even if they reside on a different subnet.*

## Protected Member Functions

- [IManagedBusManager](#) ()

## Static Package Functions

- static void [ConvertToNativeGuid](#) ([ManagedPGRGuid](#)^ mgdPGRGuid, FlyCapture2::PGRGuid \*pgrGuid)  
Convert a [ManagedPGRGuid](#) to a native PGRGuid.
- static void [ConvertToManagedGuid](#) (FlyCapture2::PGRGuid \*pgrGuid, [ManagedPGRGuid](#)^ mgdPGRGuid)  
Convert a native PGRGuid to a [ManagedPGRGuid](#).

### 8.26.1 Detailed Description

[ManagedBusManager](#) provides the functionality for the user to get an PGRGuid for a desired camera or device easily.

Once the camera or device token is found, it can then be used to connect to the camera or device through the camera class or device class. In addition, the BusManager class provides the ability to be notified when a camera or device is added or removed or some event occurs on the PC.

### 8.26.2 Constructor & Destructor Documentation

#### 8.26.2.1 ManagedBusManager ( )

#### 8.26.2.2 ~ManagedBusManager ( )

#### 8.26.2.3 !ManagedBusManager ( ) [protected]

### 8.26.3 Member Function Documentation

#### 8.26.3.1 void ConvertToManagedGuid ( FlyCapture2::PGRGuid \* pgrGuid, ManagedPGRGuid^ mgdPGRGuid ) [inline, static, package]

Convert a native PGRGuid to a [ManagedPGRGuid](#).

##### Parameters

<i>pgrGuid</i>	The native PGRGuid.
<i>mgdPGR-Guid</i>	The <a href="#">ManagedPGRGuid</a> .

#### 8.26.3.2 void ConvertToNativeGuid ( ManagedPGRGuid^ mgdPGRGuid, FlyCapture2::PGRGuid \* pgrGuid ) [inline, static, package]

Convert a [ManagedPGRGuid](#) to a native PGRGuid.



## Parameters

<i>mgdPGR-Guid</i>	The <a href="#">ManagedPGRGuid</a> .
<i>pgrGuid</i>	The native PGRGuid.

8.26.3.3 `array< CameraInfo^ > DiscoverGigECameras ( ) [static]`

Discover all cameras connected to the network even if they reside on a different subnet.

This is useful in situations where a GigE camera is using Persistent IP and the application's subnet is different from the device subnet. After discovering the camera, it is easy to use [ForceIPAddressToCamera\(\)](#) to set a different IP configuration.

## Returns

Array of [CameraInfo](#) structures containing information about discovered cameras.

8.26.3.4 `void FireBusReset ( ManagedPGRGuid^ guid )`

Fire a bus reset.

The actual bus reset is only fired for the specified 1394 bus, but it will effectively cause a global bus reset for the library.

## Parameters

<i>guid</i>	<a href="#">ManagedPGRGuid</a> of the camera or the device to cause bus reset.
-------------	--

8.26.3.5 `void ForceAllIPAddressesAutomatically ( ) [static]`

Force all cameras on the network to be assigned sequential IP addresses on the same subnet as the network adapters that they are connected to.

This is useful in situations where a GigE Vision cameras are using Persistent IP addresses and the application's subnet is different from the devices.

8.26.3.6 `void ForceAllIPAddressesAutomatically ( unsigned int serialNumber ) [static]`

Force a specific cameras on the network to be assigned sequential IP address on the same subnet as the network adapters that it is connected to.

This is useful in situations where a GigE Vision camera is using Persistent IP addresses and the application's subnet is different from the device.

8.26.3.7 void ForceIPAddressToCamera ( System::Net::NetworkInformation::PhysicalAddress^ *macAddress*, System::Net::IPAddress^ *ipAddress*, System::Net::IPAddress^ *subnetMask*, System::Net::IPAddress^ *defaultGateway* ) [static]

Force the camera with the specific MAC address to the specified IP address, subnet mask and default gateway.

This is useful in situations where a GigE Vision camera is using Persistent IP and the application's subnet is different from the device subnet.

#### Parameters

<i>macAddress</i>	MAC address of the camera.
<i>ipAddress</i>	IP address to set on the camera.
<i>subnetMask</i>	Subnet mask to set on the camera.
<i>default-Gateway</i>	Default gateway to set on the camera.

8.26.3.8 ManagedPGRGuid GetCameraFromIndex ( unsigned int *index* )

Gets the [ManagedPGRGuid](#) for a camera on the PC.

It uniquely identifies the camera specified by the index and is used to identify the camera during a [ManagedCamera::Connect\(\)](#) call.

#### Parameters

<i>index</i>	Zero based index of camera.
--------------	-----------------------------

#### Returns

Unique [ManagedPGRGuid](#) for the camera.

8.26.3.9 ManagedPGRGuid GetCameraFromIPAddress ( System::Net::IPAddress^ *ipAddress* )

Gets the [ManagedPGRGuid](#) for a camera with the specified IPv4 address.

#### Parameters

<i>ipAddress</i>	IP address to get <a href="#">ManagedPGRGuid</a> for.
------------------	---

#### Returns

Unique [ManagedPGRGuid](#) for the camera.

**8.26.3.10 ManagedPGRGuid GetCameraFromSerialNumber ( unsigned int *serialNumber* )**

Gets the [ManagedPGRGuid](#) for a camera on the PC.

It uniquely identifies the camera specified by the serial number and is used to identify the camera during a [ManagedCamera::Connect\(\)](#) call.

**Parameters**

<i>serial- Number</i>	Serial number of camera.
---------------------------	--------------------------

**See also**

[GetCameraFromIndex\(\)](#)

**Returns**

Unique [ManagedPGRGuid](#) for the camera.

**8.26.3.11 unsigned int GetCameraSerialNumberFromIndex ( unsigned int *index* )**

Gets the serial number of the camera with the specified index.

**Parameters**

<i>index</i>	Zero based index of desired camera.
--------------	-------------------------------------

**Returns**

Serial number of camera.

**8.26.3.12 ManagedPGRGuid GetDeviceFromIndex ( unsigned int *index* )**

Gets the [ManagedPGRGuid](#) for a device.

It uniquely identifies the device specified by the index.

**Parameters**

<i>index</i>	Zero based index of device.
--------------	-----------------------------

**See also**

[GetNumOfDevices\(\)](#)

**Returns**

Unique [ManagedPGRGuid](#) for the device.

**8.26.3.13 InterfaceType GetInterfaceTypeFromGuid ( ManagedPGRGuid<sup>^</sup> guid )**

Gets the interface type associated with a [ManagedPGRGuid](#).

This is useful in situations where there is a need to enumerate all cameras for a particular interface.

**Parameters**

<i>guid</i>	The <a href="#">ManagedPGRGuid</a> to get the interface for.
-------------	--

**Returns**

The interface type of the PGRGuid.

**8.26.3.14 unsigned int GetNumOfCameras ( )**

Gets the number of cameras attached to the PC.

**Returns**

The number of cameras attached.

**8.26.3.15 unsigned int GetNumOfDevices ( )**

Gets the number of devices.

This may include hubs, host controllers and other hardware devices (including cameras).

**Returns**

The number of devices found.

**8.26.3.16 ManagedTopologyNode GetTopology ( )**

Gets the topology information for the PC.

**Returns**

[ManagedTopologyNode](#) object that will contain the topology

**8.26.3.17** unsigned int GetUsbLinkInfo ( ManagedPGRGuid^ *guid* )

Read usb link info for the port that the specified device is connected to.

**Parameters**

<i>guid</i>	PGRGuid of the device to read from.
-------------	-------------------------------------

**Returns**

Value read from the card register.

**8.26.3.18** unsigned int GetUsbPortStatus ( ManagedPGRGuid^ *guid* )

Read usb port status for the port that the specified device is connected to.

**Parameters**

<i>guid</i>	PGRGuid of the device to read from.
-------------	-------------------------------------

**Returns**

Value read from the card register.

**8.26.3.19** bool IsCameraControlable ( ManagedPGRGuid^ *guid* )

Query CCP status on camera with corresponding PGRGuid.

This is useful to determine if a GigE camera can be controlled.

**Parameters**

<i>pGuid</i>	PGRGuid of the camera
--------------	-----------------------

**Returns**

True means camera is controlable.

**8.26.3.20** unsigned int ReadPhyRegister ( ManagedPGRGuid^ *guid*, unsigned int *page*, unsigned int *port*, unsigned int *address* )

Read a phy register on the specified device.

The full address to be read from is determined by the page, port and address.

## Parameters

<i>guid</i>	<a href="#">ManagedPGRGuid</a> of the device to read from.
<i>page</i>	Page to read from.
<i>port</i>	Port to read from.
<i>address</i>	Address to read from.

## Returns

Value read from the phy register.

### 8.26.3.21 `System::IntPtr RegisterCallback ( EnumCallback^ hCallbackDelegate, ManagedCallbackType callbackType, System::IntPtr parameter )`

Register a callback function that will be called when the specified callback event occurs.

## Parameters

<i>hCallback-Delegate</i>	Handle to EnumCallback function to receive the callback.
<i>callbackType</i>	Type of callback to register for.
<i>parameter</i>	Callback parameter to be passed to callback.

## See also

[UnregisterCallback\(\)](#)

## Returns

Unique callback handle used for unregistering callback.

### 8.26.3.22 `void RescanBus ( )`

Force a rescan of the buses.

This does not trigger a bus reset. The camera objects will be invalidated only if the camera network topology is changed (ie. a camera is disconnected or added)

### 8.26.3.23 `void UnregisterCallback ( System::IntPtr callbackHandle )`

Unregister a callback function.

## Parameters

<i>callback-Handle</i>	Unique callback handle.
------------------------	-------------------------

See also

[RegisterCallback\(\)](#)

8.26.3.24 void WritePhyRegister ( ManagedPGRGuid^ *guid*, unsigned int *page*, unsigned int *port*, unsigned int *address*, unsigned int *regVal* )

Write a phy register on the specified device.  
The full address to be written to is determined by the page, port and address.

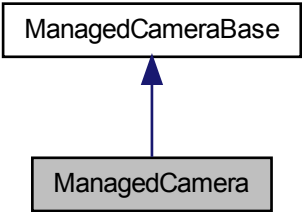
Parameters

<i>guid</i>	<a href="#">ManagedPGRGuid</a> of the device to write to.
<i>page</i>	Page to write to.
<i>port</i>	Port to write to.
<i>address</i>	Address to write to.
<i>regVal</i>	Value to write to phy register.

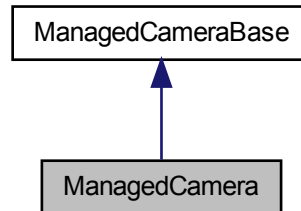
8.27 ManagedCamera Class Reference

[ManagedCamera](#) represents a physical camera that uses the IIDC register set.

Inheritance diagram for ManagedCamera:



Collaboration diagram for ManagedCamera:



### Public Member Functions

- [ManagedCamera](#) ()
- [~ManagedCamera](#) ()
- virtual void [Connect](#) ([ManagedPGRGuid](#)<sup>^</sup> mgdPGRGuid) override  
*Connects the [ManagedCamera](#) object to the camera specified by the GUID.*

### Protected Member Functions

- [!ManagedCamera](#) ()

### DCAM Formats

These functions deal with DCAM video mode and frame rate on the camera.

They are only used for firewire and Usb2 cameras.

- bool [GetVideoModeAndFrameRateInfo](#) ([VideoMode](#) videoMode, [FrameRate](#) frameRate)  
*Query the camera to determine if the specified video mode and frame rate is supported.*
- void [GetVideoModeAndFrameRate](#) ([VideoMode](#)% videoMode, [FrameRate](#)% frameRate)  
*Get the current video mode and frame rate from the camera.*
- void [SetVideoModeAndFrameRate](#) ([VideoMode](#) videoMode, [FrameRate](#) frameRate)  
*Set the specified video mode and frame rate to the camera.*



## Format7

These functions deal with Format7 custom image control on the camera.

- [Format7Info](#)<sup>^</sup> [GetFormat7Info](#) ([Mode](#) mode, bool% supported)  
*Retrieve the availability of Format7 custom image mode and the camera capabilities for the specified Format7 mode.*
- [Format7PacketInfo](#)<sup>^</sup> [ValidateFormat7Settings](#) ([Format7ImageSettings](#)<sup>^</sup> imageSettings, bool% settingsAreValid)  
*Validates [Format7ImageSettings](#) structure and returns valid packet size information if the image settings are valid.*
- void [GetFormat7Configuration](#) ([Format7ImageSettings](#)<sup>^</sup> imageSettings, unsigned int% packetSize, float% percentSpeed)  
*Get the current Format7 configuration from the camera.*
- void [SetFormat7Configuration](#) ([Format7ImageSettings](#)<sup>^</sup> imageSettings, unsigned int recommendedPacketSize)  
*Set the current Format7 configuration to the camera.*
- void [SetFormat7Configuration](#) ([Format7ImageSettings](#)<sup>^</sup> imageSettings, float recommendedPercentSpeed)  
*Set the current Format7 configuration to the camera.*
- static void [StartSyncCapture](#) (unsigned int numCameras, array< [ManagedCamera](#)<sup>^</sup> >^ppCameras)  
*Start multiple cameras in synchronization.*
- static void [StartSyncCapture](#) (unsigned int numCameras, array< [ManagedCamera](#)<sup>^</sup> >^ppCameras, array< [ImageEventCallback](#)<sup>^</sup> >^pCallbackFns, array< [IntPtr](#)<sup>^</sup> >^pCallbackDataArray)  
*Start multiple cameras in synchronization using callbacks.*

### 8.27.1 Detailed Description

[ManagedCamera](#) represents a physical camera that uses the IIDC register set.

The object must first be connected to using [Connect\(\)](#) before any other operations can proceed.

It is possible for more than 1 Camera object to connect to a single physical camera. However, isochronous transmission to more than 1 Camera object is not supported.

### 8.27.2 Constructor & Destructor Documentation

#### 8.27.2.1 [ManagedCamera](#) ( )

#### 8.27.2.2 [~ManagedCamera](#) ( )

#### 8.27.2.3 [!ManagedCamera](#) ( ) [protected]

### 8.27.3 Member Function Documentation

**8.27.3.1** `void Connect ( ManagedPGRGuid^ mgdPGRGuid )` [override, virtual]

Connects the [ManagedCamera](#) object to the camera specified by the GUID.

#### Parameters

<i>mgdPGR-Guid</i>	The unique identifier for a specific camera on the PC.
--------------------	--

#### See also

[ManagedBusManager::GetCameraFromIndex\(\)](#)  
[ManagedBusManager::GetCameraFromSerialNumber\(\)](#)

Reimplemented from [ManagedCameraBase](#).

**8.27.3.2** `void GetFormat7Configuration ( Format7ImageSettings^ imageSettings, unsigned int% packetSize, float% percentSpeed )`

Get the current Format7 configuration from the camera.

This call will only succeed if the camera is already in Format7.

#### Parameters

<i>image-Settings</i>	Current image settings.
<i>packetSize</i>	Current packet size.
<i>percent-Speed</i>	Current packet size as a percentage.

#### See also

[GetFormat7Info\(\)](#)  
[ValidateFormat7Settings\(\)](#)  
[SetFormat7Configuration\(\)](#)  
[GetVideoModeAndFrameRate\(\)](#)

**8.27.3.3** `Format7Info GetFormat7Info ( Mode mode, bool% supported )`

Retrieve the availability of Format7 custom image mode and the camera capabilities for the specified Format7 mode.

The mode must be specified in the [Format7Info](#) structure in order for the function to succeed.

## Parameters

<i>mode</i>	Format7 mode to query.
<i>supported</i>	Whether the specified mode is supported.

## See also

[ValidateFormat7Settings\(\)](#)  
[GetFormat7Configuration\(\)](#)  
[SetFormat7Configuration\(\)](#)

## Returns

[Format7Info](#) structure filled with the capabilities of the specified mode and the current state in the specified mode.

**8.27.3.4** void [GetVideoModeAndFrameRate](#) ( **VideoMode%** *videoMode*, **FrameRate%** *frameRate* )

Get the current video mode and frame rate from the camera.

If the camera is in Format7, the video mode will be VIDEOMODE\_FORMAT7 and the frame rate will be FRAMERATE\_FORMAT7.

## Parameters

<i>videoMode</i>	Current video mode.
<i>frameRate</i>	Current frame rate.

## See also

[GetVideoModeAndFrameRateInfo\(\)](#)  
[SetVideoModeAndFrameRate\(\)](#)

**8.27.3.5** bool [GetVideoModeAndFrameRateInfo](#) ( **VideoMode** *videoMode*, **FrameRate** *frameRate* )

Query the camera to determine if the specified video mode and frame rate is supported.

## Parameters

<i>videoMode</i>	Video mode to check.
<i>frameRate</i>	Frame rate to check.

## See also

[GetVideoModeAndFrameRate\(\)](#)  
[SetVideoModeAndFrameRate\(\)](#)

**Returns**

Whether the video mode and frame rate is supported.

**8.27.3.6** `void SetFormat7Configuration ( Format7ImageSettings^ imageSettings, unsigned int recommendedPacketSize )`

Set the current Format7 configuration to the camera.

**Parameters**

<i>image-Settings</i>	Image settings to be written to the camera.
<i>recommended-PacketSize</i>	Packet size to be written to the camera.

**See also**

[GetFormat7Info\(\)](#)  
[ValidateFormat7Settings\(\)](#)  
[GetFormat7Configuration\(\)](#)

**8.27.3.7** `void SetFormat7Configuration ( Format7ImageSettings^ imageSettings, float recommendedPercentSpeed )`

Set the current Format7 configuration to the camera.

**Parameters**

<i>image-Settings</i>	Image settings to be written to the camera.
<i>recommended-Percent-Speed</i>	Percentage of packet size to be written to the camera.

**See also**

[GetFormat7Info\(\)](#)  
[ValidateFormat7Settings\(\)](#)  
[GetFormat7Configuration\(\)](#)

**8.27.3.8** `void SetVideoModeAndFrameRate ( VideoMode videoMode, FrameRate frameRate )`

Set the specified video mode and frame rate to the camera.

It is not possible to set the camera to VIDEOMODE\_FORMAT7 or FRAMERATE\_FORMAT7. Use the Format7 functions to set the camera into Format7.

## Parameters

<i>videoMode</i>	Video mode to set to camera.
<i>frameRate</i>	Frame rate to set to camera.

## See also

[GetVideoModeAndFrameRateInfo\(\)](#)  
[GetVideoModeAndFrameRate\(\)](#)

**8.27.3.9** `void StartSyncCapture ( unsigned int numCameras, array< ManagedCamera^ >^ ppCameras ) [static]`

Start multiple cameras in synchronization.

This function is only used for firewire cameras.

## Parameters

<i>num-Cameras</i>	Number of cameras to start.
<i>ppCameras</i>	An array of <a href="#">ManagedCamera</a> objects to be started.

## See also

[StartCapture\(\)](#)

**8.27.3.10** `void StartSyncCapture ( unsigned int numCameras, array< ManagedCamera^ >^ ppCameras, array< ImageEventCallback^ >^ pCallbackFns, array< IntPtr^ >^ pCallbackDataArray ) [static]`

Start multiple cameras in synchronization using callbacks.

This function is only used for firewire cameras.

## Parameters

<i>num-Cameras</i>	Number of cameras to start.
<i>ppCameras</i>	An array of <a href="#">ManagedCamera</a> objects to be started
<i>pCallback-Fns</i>	An array of callback functions
<i>pCallback-DataArray</i>	An array of <a href="#">ManagedImage</a> objects to be populated during callback

See also

[StartCapture\(\)](#)

#### 8.27.3.11 **Format7PacketInfo** ValidateFormat7Settings ( **Format7ImageSettings**^ *imageSettings*, bool% *settingsAreValid* )

Validates [Format7ImageSettings](#) structure and returns valid packet size information if the image settings are valid.

The current image settings are cached while validation is taking place. The cached settings are restored when validation is complete.

##### Parameters

<i>image-Settings</i>	Structure containing the image settings.
<i>settingsAre-Valid</i>	Whether the settings are valid.

See also

[GetFormat7Info\(\)](#)  
[GetFormat7Configuration\(\)](#)  
[SetFormat7Configuration\(\)](#)

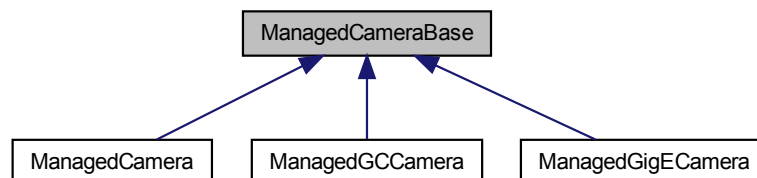
##### Returns

Packet size information that can be used to determine a valid packet size.

## 8.28 ManagedCameraBase Class Reference

Abstract base class that represents a generic camera that defines a general interface to a camera.

Inheritance diagram for ManagedCameraBase:



## Public Member Functions

- virtual [~ManagedCameraBase](#) ()
- void [SetCamera](#) (System::IntPtr otherCamera)  
*Set camera from a integer pointer camera.*
- virtual [TimeStamp](#)^ [GetCycleTime](#) ()  
*Returns a Timestamp struct containing 1394 CYCLE\_TIME information.*
- virtual [CameraStats](#)^ [GetStats](#) ()
- virtual void [ResetStats](#) ()
- virtual void [RegisterEvent](#) ([ManagedEventOptions](#)^ hOpts)
- virtual void [DeregisterEvent](#) ([ManagedEventOptions](#)^ hOpts)
- virtual void [RegisterAllEvents](#) ([ManagedEventOptions](#)^ hOpts)
- virtual void [DeregisterAllEvents](#) ()

## Connection and Image Retrieval

*These functions deal with connections and image retrieval from the camera.*

- virtual void [Connect](#) ([ManagedPGRGuid](#)^ mgdPGRGuid)  
*Connects the [ManagedCamera](#) object to the camera specified by the GUID.*
- virtual void [Disconnect](#) ()  
*Disconnects the [ManagedCamera](#) object from the camera.*
- virtual bool [IsConnected](#) ()  
*Checks if the [ManagedCamera](#) object is connected to a physical camera specified by a GUID.*
- virtual void [SetCallback](#) ([ImageEventCallback](#)^ hCallbackDelegate)  
*Sets the callback data to be used on completion of image transfer.*
- virtual void [StartCapture](#) ()  
*Starts isochronous image capture.*
- virtual void [StartCapture](#) ([ImageEventCallback](#)^ hCallbackDelegate)  
*Starts isochronous image capture.*
- virtual void [StopCapture](#) ()  
*Stops isochronous image transfer and cleans up all associated resources.*
- virtual void [RetrieveBuffer](#) ([ManagedImage](#)^ image)  
*Retrieves the the next image object containing the next image.*
- virtual void [WaitForBufferEvent](#) ([ManagedImage](#)^ image, unsigned int event-Number)  
*Retrieves the next image event containing the next part of the image.*
- virtual void [SetUserBuffers](#) (IntPtr pMemBuffers, int size, int numBuffers)  
*Specify user allocated buffers to use as image data buffers.*
- virtual [FC2Config](#)^ [GetConfiguration](#) ()  
*Get the configuration associated with the camera object.*
- virtual void [SetConfiguration](#) ([FC2Config](#)^ config)  
*Set the configuration associated with the camera object.*

### Information and Properties

These functions deal with information and properties can be retrieved from the camera.

- virtual [CameraInfo](#)^ [GetCameraInfo](#) ()  
Retrieves information from the camera such as serial number, model name and other camera information.
- virtual [CameraPropertyInfo](#)^ [GetPropertyInfo](#) ([PropertyType](#) type)  
Retrieves information about the specified camera property.
- virtual [CameraProperty](#)^ [GetProperty](#) ([PropertyType](#) type)  
Reads the settings for the specified property from the camera.
- virtual void [SetProperty](#) ([CameraProperty](#)^ camProperty)  
Writes the settings for the specified property to the camera.
- virtual void [SetProperty](#) ([CameraProperty](#)^ camProperty, bool broadcast)  
Writes the settings for the specified property to the camera.

### General Purpose Input / Output

These functions deal with general GPIO pin control on the camera.

- virtual unsigned int [GetGPIOPinDirection](#) (unsigned int pin)  
Get the GPIO pin direction for the specified pin.
- virtual void [SetGPIOPinDirection](#) (unsigned int pin, unsigned int direction)  
Set the GPIO pin direction for the specified pin.
- virtual void [SetGPIOPinDirection](#) (unsigned int pin, unsigned int direction, bool broadcast)  
Set the GPIO pin direction for the specified pin.

### Trigger

These functions deal with trigger control on the camera.

- virtual [TriggerModelInfo](#)^ [GetTriggerModelInfo](#) ()  
Retrieve trigger information from the camera.
- virtual [TriggerMode](#)^ [GetTriggerMode](#) ()  
Retrieve current trigger settings from the camera.
- virtual void [SetTriggerMode](#) ([TriggerMode](#)^ triggerMode)  
Set the specified trigger settings to the camera.
- virtual void [FireSoftwareTrigger](#) (bool broadcast)  
Fire the software trigger according to the DCAM specifications.
- virtual [CameraPropertyInfo](#)^ [GetTriggerDelayInfo](#) ()  
Retrieve trigger delay information from the camera.
- virtual [CameraProperty](#)^ [GetTriggerDelay](#) ()  
Retrieve current trigger delay settings from the camera.
- virtual void [SetTriggerDelay](#) ([CameraProperty](#)^ triggerDelay)  
Set the specified trigger delay settings to the camera.
- virtual void [SetTriggerDelay](#) ([CameraProperty](#)^ triggerDelay, bool broadcast)  
Set the specified trigger delay settings to the camera.



### Strobe

*These functions deal with strobe control on the camera.*

- virtual [StrobeInfo](#)^ [GetStrobeInfo](#) (unsigned int source)  
*Retrieve strobe information from the camera.*
- virtual [StrobeControl](#)^ [GetStrobe](#) (unsigned int source)  
*Retrieve current strobe settings from the camera.*
- virtual void [SetStrobe](#) ([StrobeControl](#)^ strobeControl)  
*Set current strobe settings to the camera.*

### Look Up Table

*These functions deal with Look Up Table control on the camera.*

*Note that some cameras may report support for the LUT and return an inputBitDepth of 0. In these cases use  $\log_2(\text{numEntries})$  for the inputBitDepth.*

- virtual [LutData](#)^ [GetLUTInfo](#) ()  
*Query if LUT support is available on the camera.*
- virtual void [GetLUTBankInfo](#) (unsigned int bank, bool% readSupported, bool% writeSupported)  
*Query the read/write status of a single LUT bank.*
- virtual unsigned int [GetActiveLUTBank](#) ()  
*Get the LUT bank that is currently being used.*
- virtual void [SetActiveLUTBank](#) (unsigned int activeBank)  
*Set the LUT bank that will be used.*
- virtual void [EnableLUT](#) (bool on)  
*Enable or disable LUT functionality on the camera.*
- virtual void [GetLUTChannel](#) (unsigned int bank, unsigned int channel, unsigned int sizeEntries, array< unsigned int >^entries)  
*Get the LUT channel settings from the camera.*
- virtual void [SetLUTChannel](#) (unsigned int bank, unsigned int channel, unsigned int sizeEntries, array< unsigned int >^entries)  
*Set the LUT channel settings to the camera.*

### Memory Channels

*These functions deal with memory channel control on the camera.*

- virtual unsigned int [GetMemoryChannel](#) ()  
*Retrieve the current memory channel from the camera.*
- virtual void [SaveToMemoryChannel](#) (unsigned int channel)  
*Save the current settings to the specified current memory channel.*
- virtual void [RestoreFromMemoryChannel](#) (unsigned int channel)  
*Restore the specified current memory channel.*
- virtual unsigned int [GetMemoryChannelInfo](#) ()  
*Query the camera for memory channel support.*

### Embedded Image Information

*These functions deal with embedded image information control on the camera.*

- virtual [EmbeddedImageInfo](#)^ [GetEmbeddedImageInfo](#) ()  
*Get the current status of the embedded image information register, as well as the availability of each embedded property.*
- virtual void [SetEmbeddedImageInfo](#) ([EmbeddedImageInfo](#)^ info)  
*Sets the on/off values of the embedded image information structure to the camera.*

### Register Operation

*These functions deal with register operation on the camera.*

- virtual void [WriteRegister](#) (unsigned int address, unsigned int value)  
*Write to the specified register on the camera.*
- virtual void [WriteRegister](#) (unsigned int address, unsigned int value, bool broadcast)  
*Write to the specified register on the camera.*
- virtual unsigned int [ReadRegister](#) (unsigned int address)  
*Read the specified register from the camera.*
- virtual void [WriteRegisterBlock](#) (unsigned short addressHigh, unsigned int addressLow, array< unsigned int >^buffer)  
*Write to the specified register block on the camera.*
- virtual void [ReadRegisterBlock](#) (unsigned short addressHigh, unsigned int addressLow, array< unsigned int >^buffer)  
*Read from the specified register block on the camera.*

### Static Public Member Functions

- static System::String^ [GetRegisterString](#) (unsigned int registerVal)  
*Returns a text representation of the register value.*

### Protected Member Functions

- [ManagedCameraBase](#) ()
- void [OnNativeCallback](#) (FlyCapture2::Image \*pImage, void \*pCallbackData)
- void [OnNativeCameraEventCallback](#) (void \*pCallbackData)

### Protected Attributes

- FlyCapture2::CameraBase \* [m\\_pNativeCamBase](#)
- bool [m\\_isLocal](#)
- ImageEventCallback^ [m\\_externalDelegate](#)
- ImageCallbackDelegate^ [m\\_internalDelegate](#)
- ManagedCameraEventCallbackDelegate^ [m\\_internalCameraEventDelegate](#)
- IntPtr [m\\_p](#)

- Dictionary < [ManagedEventOptions](#)<sup>^</sup> , [NativeEventStruct](#) ><sup>^</sup> [m\\_specific-InternalCameraEvents](#)
- Dictionary < [ManagedEventOptions](#)<sup>^</sup> , [NativeEventStruct](#) ><sup>^</sup> [m\\_allInternal-CameraEvents](#)

## Package Functions

- [FlyCapture2::CameraBase](#) \* [GetNativeCamera](#) ()

### 8.28.1 Detailed Description

Abstract base class that represents a generic camera that defines a general interface to a camera.

### 8.28.2 Constructor & Destructor Documentation

8.28.2.1 `virtual ~ManagedCameraBase ( ) [inline, virtual]`

8.28.2.2 `ManagedCameraBase ( ) [inline, protected]`

### 8.28.3 Member Function Documentation

8.28.3.1 `void Connect ( ManagedPGRGuid^ mgdPGRGuid ) [virtual]`

Connects the [ManagedCamera](#) object to the camera specified by the GUID.

#### Parameters

<i>mgdPGR-Guid</i>	The unique identifier for a specific camera on the PC.
--------------------	--

#### See also

[ManagedBusManager::GetCameraFromIndex\(\)](#)  
[ManagedBusManager::GetCameraFromSerialNumber\(\)](#)

Reimplemented in [ManagedCamera](#), [ManagedGigECamera](#), and [ManagedGCCamera](#).

8.28.3.2 `void DeregisterAllEvents ( ) [virtual]`

8.28.3.3 `void DeregisterEvent ( ManagedEventOptions^ hOpts ) [virtual]`

8.28.3.4 `void Disconnect ( void ) [virtual]`

Disconnects the [ManagedCamera](#) object from the camera.

This allows another physical camera specified by a GUID to be connected to the - [ManagedCamera](#) object.

See also

[Connect\(\)](#)

Reimplemented in [ManagedGCCamera](#).

#### 8.28.3.5 void EnableLUT ( bool *on* ) [virtual]

Enable or disable LUT functionality on the camera.

Parameters

<i>on</i>	Whether to enable or disable LUT.
-----------	-----------------------------------

See also

[GetLUTInfo\(\)](#)

[GetLUTChannel\(\)](#)

[SetLUTChannel\(\)](#)

#### 8.28.3.6 void FireSoftwareTrigger ( bool *broadcast* ) [virtual]

Fire the software trigger according to the DCAM specifications.

Parameters

<i>broadcast</i>	Whether the action should be broadcast.
------------------	---

#### 8.28.3.7 unsigned int GetActiveLUTBank ( ) [virtual]

Get the LUT bank that is currently being used.

For cameras with PGR LUT, the active bank is always 0.

Returns

The currently active bank.

#### 8.28.3.8 CameraInfo GetCameraInfo ( ) [virtual]

Retrieves information from the camera such as serial number, model name and other camera information.

**Returns**

Structure containing camera information.

**8.28.3.9 FC2Config GetConfiguration ( ) [virtual]**

Get the configuration associated with the camera object.

**See also**

[SetConfiguration\(\)](#)

**Returns**

Current configuration.

**8.28.3.10 TimeStamp GetCycleTime ( ) [virtual]**

Returns a Timestamp struct containing 1394 CYCLE\_TIME information.

**Parameters**

<i>registerVal</i>	The register value to query.
--------------------	------------------------------

**Returns**

An Error indicating the success or failure of the function.

**8.28.3.11 EmbeddedImageInfo GetEmbeddedImageInfo ( ) [virtual]**

Get the current status of the embedded image information register, as well as the availability of each embedded property.

**See also**

[SetEmbeddedImageInfo\(\)](#)

**Returns**

[EmbeddedImageInfo](#) structure containing embedded image information.

**8.28.3.12 unsigned int GetGPIOPinDirection ( unsigned int *pin* ) [virtual]**

Get the GPIO pin direction for the specified pin.

This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

## Parameters

<i>pin</i>	Pin to get the direction for.
------------	-------------------------------

## See also

[SetGPIOPinDirection\(\)](#)

## Returns

Direction of the pin. 0 for input, 1 for output.

**8.28.3.13** `void GetLUTBankInfo ( unsigned int bank, bool% readSupported, bool% writeSupported )` [virtual]

Query the read/write status of a single LUT bank.

## Parameters

<i>bank</i>	The bank to query.
<i>read-Supported</i>	Whether reading from the bank is supported.
<i>write-Supported</i>	Whether writing to the bank is supported.

**8.28.3.14** `void GetLUTChannel ( unsigned int bank, unsigned int channel, unsigned int sizeEntries, array< unsigned int >^ entries )` [virtual]

Get the LUT channel settings from the camera.

## Parameters

<i>bank</i>	Bank to retrieve.
<i>channel</i>	Channel to retrieve.
<i>sizeEntries</i>	Number of entries in LUT table to read.
<i>entries</i>	Array to store LUT entries in.

## See also

[GetLUTInfo\(\)](#)  
[EnableLUT\(\)](#)  
[SetLUTChannel\(\)](#)

**8.28.3.15** `LutData GetLUTInfo ( )` [virtual]

Query if LUT support is available on the camera.

## See also

[EnableLUT\(\)](#)  
[GetLUTChannel\(\)](#)  
[SetLUTChannel\(\)](#)

## Returns

[LutData](#) structure containing the LUT information.

**8.28.3.16** `unsigned int GetMemoryChannel ( ) [virtual]`

Retrieve the current memory channel from the camera.

## See also

[SaveToMemoryChannel\(\)](#)  
[RestoreFromMemoryChannel\(\)](#)  
[GetMemoryChannelInfo\(\)](#)

## Returns

Currently selected memory channel.

**8.28.3.17** `unsigned int GetMemoryChannelInfo ( ) [virtual]`

Query the camera for memory channel support.

If the number of channels is 0, then memory channel support is not available.

## See also

[GetMemoryChannel\(\)](#)  
[SaveToMemoryChannel\(\)](#)  
[RestoreFromMemoryChannel\(\)](#)

## Returns

Number of memory channels supported.

**8.28.3.18** `FlyCapture2::CameraBase * GetNativeCamera ( ) [package]`**8.28.3.19** `CameraProperty GetProperty ( PropertyType type ) [virtual]`

Reads the settings for the specified property from the camera.

If auto is on, the integer and abs values returned may not be consistent with each other.

## Parameters

<i>type</i>	The PropertyType to retrieve information about.
-------------	---

## See also

[GetPropertyInfo\(\)](#)[SetProperty\(\)](#)

## Returns

Property structure containing property information.

**8.28.3.20 CameraPropertyInfo GetPropertyInfo ( PropertyType *type* )** [virtual]

Retrieves information about the specified camera property.

## Parameters

<i>type</i>	The PropertyType to retrieve information about.
-------------	---

## See also

[GetProperty\(\)](#)[SetProperty\(\)](#)

## Returns

PropertyInfo structure containing property information.

**8.28.3.21 System::String GetRegisterString ( unsigned int *registerVal* )** [static]

Returns a text representation of the register value.

## Parameters

<i>registerVal</i>	The register value to query.
--------------------	------------------------------

## Returns

The text representation of the register.

**8.28.3.22 CameraStats GetStats ( )** [virtual]



**8.28.3.23 StrobeControl GetStrobe ( unsigned int *source* )** [virtual]

Retrieve current strobe settings from the camera.

**Parameters**

<i>source</i>	Source pin for strobe information.
---------------	------------------------------------

**See also**

[GetStrobeInfo\(\)](#)  
[SetStrobe\(\)](#)

**Returns**

[StrobeControl](#) structure containing strobe information.

**8.28.3.24 StrobeInfo GetStrobeInfo ( unsigned int *source* )** [virtual]

Retrieve strobe information from the camera.

**Parameters**

<i>source</i>	Source pin for strobe information.
---------------	------------------------------------

**See also**

[GetStrobe\(\)](#)  
[SetStrobe\(\)](#)

**Returns**

[StrobeInfo](#) structure containing strobe information.

**8.28.3.25 CameraProperty GetTriggerDelay ( )** [virtual]

Retrieve current trigger delay settings from the camera.

**See also**

[GetTriggerModelInfo\(\)](#)  
[GetTriggerMode\(\)](#)  
[SetTriggerMode\(\)](#)  
[GetTriggerDelayInfo\(\)](#)  
[SetTriggerDelay\(\)](#)

**Returns**

Structure to receive trigger delay settings.

**8.28.3.26 CameraPropertyInfo GetTriggerDelayInfo ( ) [virtual]**

Retrieve trigger delay information from the camera.

**See also**

[GetTriggerModelInfo\(\)](#)  
[GetTriggerMode\(\)](#)  
[SetTriggerMode\(\)](#)  
[GetTriggerDelay\(\)](#)  
[SetTriggerDelay\(\)](#)

**Returns**

Structure to receive trigger delay information.

**8.28.3.27 TriggerMode GetTriggerMode ( ) [virtual]**

Retrieve current trigger settings from the camera.

**See also**

[GetTriggerModelInfo\(\)](#)  
[SetTriggerMode\(\)](#)

**Returns**

[TriggerMode](#) structure containing trigger mode settings.

**8.28.3.28 TriggerModelInfo GetTriggerModelInfo ( ) [virtual]**

Retrieve trigger information from the camera.

**See also**

[GetTriggerMode\(\)](#)  
[SetTriggerMode\(\)](#)

**Returns**

[TriggerModelInfo](#) structure containing receive trigger information.

**8.28.3.29** `bool IsConnected ( ) [virtual]`

Checks if the [ManagedCamera](#) object is connected to a physical camera specified by a GUID.

See also

[Connect\(\)](#)  
[Disconnect\(\)](#)

Returns

Whether [Connect\(\)](#) was called on the [ManagedCamera](#) object.

**8.28.3.30** `void OnNativeCallback ( FlyCapture2::Image * pImage, void * pCallbackData ) [protected]`**8.28.3.31** `void OnNativeCameraEventCallback ( void * pCallbackData ) [protected]`**8.28.3.32** `unsigned int ReadRegister ( unsigned int address ) [virtual]`

Read the specified register from the camera.

Parameters

<i>address</i>	DCAM address to be read from.
----------------	-------------------------------

See also

[WriteRegister\(\)](#)

Returns

The register value that is read.

**8.28.3.33** `void ReadRegisterBlock ( unsigned short addressHigh, unsigned int addressLow, array< unsigned int >^ buffer ) [virtual]`

Read from the specified register block on the camera.

Parameters

<i>addressHigh</i>	Top 16 bits of the 48 bit absolute address to read from.
<i>addressLow</i>	Bottom 32 bits of the 48 bits absolute address to read from.
<i>buffer</i>	Array to store read data.

See also

[WriteRegisterBlock\(\)](#)

8.28.3.34 void RegisterAllEvents ( **ManagedEventOptions**^ *hOpts* ) [virtual]

8.28.3.35 void RegisterEvent ( **ManagedEventOptions**^ *hOpts* ) [virtual]

8.28.3.36 void ResetStats ( ) [virtual]

8.28.3.37 void RestoreFromMemoryChannel ( unsigned int *channel* ) [virtual]

Restore the specified current memory channel.

#### Parameters

<i>channel</i>	Memory channel to restore from.
----------------	---------------------------------

See also

[GetMemoryChannel\(\)](#)  
[SaveToMemoryChannel\(\)](#)  
[GetMemoryChannelInfo\(\)](#)

8.28.3.38 void RetrieveBuffer ( **ManagedImage**^ *image* ) [virtual]

Retrieves the the next image object containing the next image.

If the grab mode has not been set, or has been set to DROP\_FRAMES the default behavior is to requeue images for DMA if they have not been retrieved by the time the next image transfer completes. If BUFFER\_FRAMES is specified, the next image in the sequence will be retrieved. Note that for the BUFFER\_FRAMES case, if retrieval does not keep up with the DMA process, images will be lost. The default behavior is to perform DROP\_FRAMES image retrieval.

#### Parameters

<i>image</i>	<a href="#">ManagedImage</a> object to store image data.
--------------	--

See also

[StartCapture\(\)](#)  
[StopCapture\(\)](#)  
[WaitForBufferEvent\(\)](#)

8.28.3.39 void SaveToMemoryChannel ( unsigned int *channel* ) [virtual]

Save the current settings to the specified current memory channel.

Parameters

<i>channel</i>	Memory channel to save to.
----------------	----------------------------

See also

[GetMemoryChannel\(\)](#)  
[RestoreFromMemoryChannel\(\)](#)  
[GetMemoryChannelInfo\(\)](#)

8.28.3.40 void SetActiveLUTBank ( unsigned int *activeBank* ) [virtual]

Set the LUT bank that will be used.

Parameters

<i>activeBank</i>	The bank to be set as active.
-------------------	-------------------------------

8.28.3.41 void SetCallback ( ImageEventCallback^ *hCallbackDelegate* ) [virtual]

Sets the callback data to be used on completion of image transfer.

To clear the current stored callback data, pass in NULL as the argument.

Parameters

<i>hCallback-Delegate</i>	A function to be called when a new image is received.
---------------------------	---

See also

[StartCapture\(\)](#)

Returns

An Error indicating the success or failure of the function.

8.28.3.42 void SetCamera ( System::IntPtr *otherCamera* )

Set camera from a integer pointer camera.

### 8.28.3.43 void SetConfiguration ( FC2Config^ *config* ) [virtual]

Set the configuration associated with the camera object.

#### Parameters

<i>config</i>	Configuration structure to be used.
---------------	-------------------------------------

#### See also

[GetConfiguration\(\)](#)

### 8.28.3.44 void SetEmbeddedImageInfo ( EmbeddedImageInfo^ *info* ) [virtual]

Sets the on/off values of the embedded image information structure to the camera.

#### Parameters

<i>info</i>	Structure to be used.
-------------	-----------------------

#### See also

[GetEmbeddedImageInfo\(\)](#)

### 8.28.3.45 void SetGPIOPinDirection ( unsigned int *pin*, unsigned int *direction* ) [virtual]

Set the GPIO pin direction for the specified pin.

This is useful if there is a need to set the pin into an input pin (i.e. to read the voltage) off the pin without setting it as a trigger source. This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

#### Parameters

<i>pin</i>	Pin to get the direction for.
<i>direction</i>	Direction of the pin. 0 for input, 1 for output.

#### See also

[GetGPIOPinDirection\(\)](#)

### 8.28.3.46 void SetGPIOPinDirection ( unsigned int *pin*, unsigned int *direction*, bool *broadcast* ) [virtual]

Set the GPIO pin direction for the specified pin.

This is useful if there is a need to set the pin into an input pin (i.e. to read the voltage) off the pin without setting it as a trigger source. This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

## Parameters

<i>pin</i>	Pin to get the direction for.
<i>direction</i>	Direction of the pin. 0 for input, 1 for output.
<i>broadcast</i>	Whether the action should be broadcast.

## See also

[GetGPIOPinDirection\(\)](#)

**8.28.3.47** `void SetLUTChannel ( unsigned int bank, unsigned int channel, unsigned int sizeEntries, array< unsigned int >^ entries ) [virtual]`

Set the LUT channel settings to the camera.

## Parameters

<i>bank</i>	Bank to set.
<i>channel</i>	Channel to set.
<i>sizeEntries</i>	Number of entries in LUT table to write. This must be the same size as numEntries returned by GetLutInfo().
<i>entries</i>	Array containing LUT entries to write.

## See also

[GetLUTInfo\(\)](#)

[EnableLUT\(\)](#)

[GetLUTChannel\(\)](#)

**8.28.3.48** `void SetProperty ( CameraProperty^ camProperty ) [virtual]`

Writes the settings for the specified property to the camera.

The property type must be specified in the Property structure passed into the function in order for the function to succeed. The absControl flag controls whether the absolute or integer value is written to the camera.

## Parameters

<i>camProperty</i>	<a href="#">CameraProperty</a> structure to be used.
--------------------	--

See also

[GetPropertyInfo\(\)](#)  
[GetProperty\(\)](#)

**8.28.3.49** void SetProperty ( CameraProperty^ *camProperty*, bool *broadcast* )  
 [virtual]

Writes the settings for the specified property to the camera.

The property type must be specified in the Property structure passed into the function in order for the function to succeed. The absControl flag controls whether the absolute or integer value is written to the camera.

Parameters

<i>camProperty</i>	<a href="#">CameraProperty</a> structure to be used.
<i>broadcast</i>	Whether the action should be broadcast.

See also

[GetPropertyInfo\(\)](#)  
[GetProperty\(\)](#)

**8.28.3.50** void SetStrobe ( StrobeControl^ *strobeControl* ) [virtual]

Set current strobe settings to the camera.

The strobe pin must be specified in the structure before being passed in to the function.

Parameters

<i>strobe- Control</i>	Structure providing strobe settings.
----------------------------	--------------------------------------

See also

[GetStrobeInfo\(\)](#)  
[GetStrobe\(\)](#)

**8.28.3.51** void SetTriggerDelay ( CameraProperty^ *triggerDelay* ) [virtual]

Set the specified trigger delay settings to the camera.

Parameters

<i>triggerDelay</i>	Structure providing trigger delay settings.
---------------------	---



## See also

[GetTriggerModelInfo\(\)](#)  
[GetTriggerMode\(\)](#)  
[SetTriggerMode\(\)](#)  
[GetTriggerDelayInfo\(\)](#)  
[GetTriggerDelay\(\)](#)

## Returns

An Error indicating the success or failure of the function.

**8.28.3.52** void SetTriggerDelay ( CameraProperty^ *triggerDelay*, bool *broadcast* )  
[virtual]

Set the specified trigger delay settings to the camera.

## Parameters

<i>triggerDelay</i>	Structure providing trigger delay settings.
<i>broadcast</i>	Whether the action should be broadcast.

## See also

[GetTriggerModelInfo\(\)](#)  
[GetTriggerMode\(\)](#)  
[SetTriggerMode\(\)](#)  
[GetTriggerDelayInfo\(\)](#)  
[GetTriggerDelay\(\)](#)

## Returns

An Error indicating the success or failure of the function.

**8.28.3.53** void SetTriggerMode ( TriggerMode^ *triggerMode* ) [virtual]

Set the specified trigger settings to the camera.

## Parameters

<i>triggerMode</i>	Structure providing trigger mode settings.
--------------------	--

## See also

[GetTriggerModelInfo\(\)](#)  
[GetTriggerMode\(\)](#)

### 8.28.3.54 void SetUserBuffers ( IntPtr *pMemBuffers*, int *size*, int *numBuffers* ) [virtual]

Specify user allocated buffers to use as image data buffers.

To prevent image tearing, the size of each buffer should be equal to  $((\text{unsigned int})(\text{bufferSize} + \text{packetSize} - 1) / \text{packetSize}) * \text{packetSize}$ . The total size should be  $(\text{size} * \text{numBuffers})$  or larger. The packet Size that should be used differs between interfaces: Firewire: Use the Format7 packet size. Usb2: First round to Format7 packet size then round to 512 bytes. Usb3: Use a packet size of 1024 bytes. GigE: No need to do any rounding on GigE

#### Parameters

<i>pMem- Buffers</i>	Pointer to memory buffers to be written to.
<i>size</i>	The size of each buffer (in bytes).
<i>numBuffers</i>	Number of buffers in the array.

#### See also

[StartCapture\(\)](#)  
[RetrieveBuffer\(\)](#)  
[StopCapture\(\)](#)

#### Returns

An Error indicating the success or failure of the function.

### 8.28.3.55 void StartCapture ( ) [virtual]

Starts isochronous image capture.

It will use either the current video mode or the most recently set video mode of the camera. [RetrieveBuffer\(\)](#) can be called to get the image data.

#### See also

[RetrieveBuffer\(\)](#)  
[StopCapture\(\)](#)

### 8.28.3.56 void StartCapture ( ImageEventCallback^ *hCallbackDelegate* ) [virtual]

Starts isochronous image capture.

It will use either the current video mode or the most recently set video mode of the camera. The callback function parameter is called on completion of image transfer.

#### Parameters

<i>hCallback-Delegate</i>	A function to be called when a new image is received.
---------------------------	---

**See also**

[RetrieveBuffer\(\)](#)  
[StopCapture\(\)](#)

**8.28.3.57** void StopCapture ( ) [virtual]

Stops isochronous image transfer and cleans up all associated resources.

If an image callback function (specified in the [StartCapture\(\)](#) call) is currently executing, [StopCapture\(\)](#) will not return until after the callback has completed.

**See also**

[StartCapture\(\)](#)  
[RetrieveBuffer\(\)](#)

**8.28.3.58** void WaitForBufferEvent ( ManagedImage^ *image*, unsigned int *eventNumber* ) [virtual]

Retrieves the next image event containing the next part of the image.

**Parameters**

<i>image</i>	<a href="#">ManagedImage</a> object to store image data.
<i>event-Number</i>	The event number to wait for.

**See also**

[StartCapture\(\)](#)  
[RetrieveBuffer\(\)](#)  
[StopCapture\(\)](#)

**8.28.3.59** void WriteRegister ( unsigned int *address*, unsigned int *value* ) [virtual]

Write to the specified register on the camera.

**Parameters**

<i>address</i>	DCAM address to be written to.
<i>value</i>	The value to be written.

See also

[ReadRegister\(\)](#)

8.28.3.60 `void WriteRegister ( unsigned int address, unsigned int value, bool broadcast )`  
`[virtual]`

Write to the specified register on the camera.

Parameters

<i>address</i>	DCAM address to be written to.
<i>value</i>	The value to be written.
<i>broadcast</i>	Whether the action should be broadcast.

See also

[ReadRegister\(\)](#)

8.28.3.61 `void WriteRegisterBlock ( unsigned short addressHigh, unsigned int addressLow,  
array< unsigned int >^ buffer ) [virtual]`

Write to the specified register block on the camera.

Parameters

<i>addressHigh</i>	Top 16 bits of the 48 bit absolute address to write to.
<i>addressLow</i>	Bottom 32 bits of the 48 bits absolute address to write to.
<i>buffer</i>	Array containing data to be written.

See also

[ReadRegisterBlock\(\)](#)

## 8.28.4 Member Data Documentation

8.28.4.1 `Dictionary<ManagedEventOptions^, NativeEventStruct> ^`  
`m_allInternalCameraEvents` `[protected]`

8.28.4.2 `ImageEventCallback ^ m_externalDelegate` `[protected]`

8.28.4.3 `ManagedCameraEventCallbackDelegate ^ m_internalCameraEventDelegate`  
`[protected]`

8.28.4.4 `ImageCallbackDelegate ^ m_internalDelegate` `[protected]`

8.28.4.5 `bool m_isLocal` [protected]

8.28.4.6 `IntPtr m_p` [protected]

8.28.4.7 `FlyCapture2::CameraBase* m_pNativeCamBase` [protected]

8.28.4.8 `Dictionary<ManagedEventOptions^, NativeEventStruct> ^ m_specificInternalCameraEvents` [protected]

## 8.29 ManagedEventCallbackData Struct Reference

### Public Attributes

- `System::String^ EventName`  
*The event name used to register the event.*
- `UInt64 EventID`  
*The device register which EventName maps to.*
- `UInt64 EventTimestamp`  
*Timestamp indicated the time (as reported by the camera) at which the camera exposure operation completed.*

### 8.29.1 Member Data Documentation

#### 8.29.1.1 `UInt64 EventID`

The device register which EventName maps to.

Provides an alternate means of indexing into different event types.

#### 8.29.1.2 `System::String^ EventName`

The event name used to register the event.

Provided so the user knows which event triggered the callback.

#### 8.29.1.3 `UInt64 EventTimestamp`

Timestamp indicated the time (as reported by the camera) at which the camera exposure operation completed.

This can be compared with image timestamps if there is a need to map event timestamps to specific images, if applicable.

## 8.30 ManagedEventOptions Struct Reference

Options for enabling device event registration.

### Public Attributes

- ManagedCameraEventCallback ^ EventCallbackFcn  
*Callback function pointer.*
- System::String ^ EventName  
*Event name to register.*

### 8.30.1 Detailed Description

Options for enabling device event registration.

### 8.30.2 Member Data Documentation

#### 8.30.2.1 ManagedCameraEventCallback ^ EventCallbackFcn

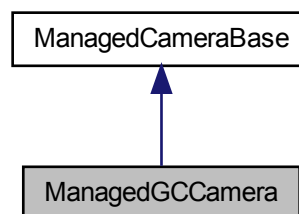
Callback function pointer.

#### 8.30.2.2 System::String ^ EventName

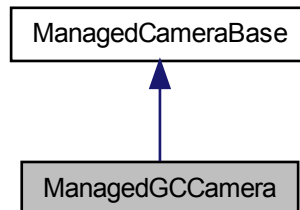
Event name to register.

## 8.31 ManagedGCCamera Class Reference

Inheritance diagram for ManagedGCCamera:



Collaboration diagram for ManagedGCCamera:



### Public Member Functions

- [ManagedGCCamera](#) (void)
- virtual [~ManagedGCCamera](#) (void)
- virtual void [Connect](#) ([ManagedPGRGuid](#)^ mgdPGRGuid) override  
*Connects the [ManagedCamera](#) object to the camera specified by the GUID.*
- virtual void [Connect](#) ([ManagedPGRGuid](#)^ mgdPGRGuid, String^ xmlPath) override
- virtual void [Disconnect](#) (void) override  
*Disconnects the [ManagedCamera](#) object from the camera.*
- void [SetCamera](#) ([ManagedCameraBase](#)^ cameraBase)
- void [SetCamera](#) ([ManagedCameraBase](#)^ cameraBase, String^ xmlPath)
- GenICam::GenApi::NodeMap^ [GetNodeMap](#) ()

### Protected Member Functions

- [!ManagedGCCamera](#) ()

### 8.31.1 Constructor & Destructor Documentation

8.31.1.1 `ManagedGCCamera ( void )`

8.31.1.2 `~ManagedGCCamera ( void )` [virtual]

8.31.1.3 `!ManagedGCCamera ( )` [protected]

### 8.31.2 Member Function Documentation

8.31.2.1 `void Connect ( ManagedPGRGuid^ mgdPGRGuid )` [override, virtual]

Connects the [ManagedCamera](#) object to the camera specified by the GUID.

#### Parameters

<i>mgdPGR-Guid</i>	The unique identifier for a specific camera on the PC.
--------------------	--

#### See also

[ManagedBusManager::GetCameraFromIndex\(\)](#)  
[ManagedBusManager::GetCameraFromSerialNumber\(\)](#)

Reimplemented from [ManagedCameraBase](#).

8.31.2.2 `void Connect ( ManagedPGRGuid^ mgdPGRGuid, String^ xmlPath )` [override, virtual]

8.31.2.3 `void Disconnect ( void )` [override, virtual]

Disconnects the [ManagedCamera](#) object from the camera.

This allows another physical camera specified by a GUID to be connected to the - [ManagedCamera](#) object.

#### See also

[Connect\(\)](#)

Reimplemented from [ManagedCameraBase](#).

8.31.2.4 `GenICam::GenApi::NodeMap GetNodeMap ( )`

8.31.2.5 `void SetCamera ( ManagedCameraBase^ cameraBase )`

8.31.2.6 `void SetCamera ( ManagedCameraBase^ cameraBase, String^ xmlPath )`

## 8.32 ManagedGCPort Class Reference

### Public Member Functions

- [ManagedGCPort](#) (GCCamera \*camera)
- virtual [~ManagedGCPort](#) (void)
- virtual void [Read](#) (IntPtr buffer, \_\_int64 address, \_\_int64 length) override
- virtual void [Write](#) (IntPtr buffer, \_\_int64 address, \_\_int64 length) override



### 8.32.1 Constructor & Destructor Documentation

8.32.1.1 **ManagedGCPort** ( GCCamera \* *camera* )

8.32.1.2 **~ManagedGCPort** ( void ) [virtual]

### 8.32.2 Member Function Documentation

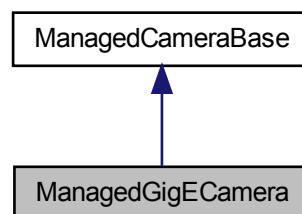
8.32.2.1 **void Read** ( IntPtr *buffer*, \_\_int64 *address*, \_\_int64 *length* ) [override, virtual]

8.32.2.2 **void Write** ( IntPtr *buffer*, \_\_int64 *address*, \_\_int64 *length* ) [override, virtual]

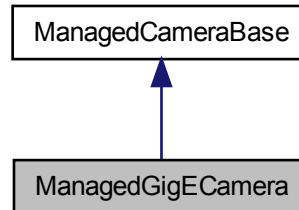
## 8.33 ManagedGigECamera Class Reference

The GigECamera object represents a physical Gigabit Ethernet camera.

Inheritance diagram for ManagedGigECamera:



Collaboration diagram for ManagedGigECamera:



### Public Member Functions

- [ManagedGigECamera](#) ()
- [~ManagedGigECamera](#) ()
- virtual void [Connect](#) ([ManagedPGRGuid](#)<sup>^</sup> mgdPGRGuid) override  
Connects the [ManagedCamera](#) object to the camera specified by the GUID.

### Protected Member Functions

- [!ManagedGigECamera](#) ()

### GVCP Register Operation

These functions deal with GVCP register operation on the camera.

- void [WriteGVCPRegister](#) (unsigned int address, unsigned int value)  
*Write a GVCP register.*
- void [WriteGVCPRegister](#) (unsigned int address, unsigned int value, bool broadcast)  
*Write a GVCP register.*
- unsigned int [ReadGVCPRegister](#) (unsigned int address)  
*Read a GVCP register.*
- void [WriteGVCPRegisterBlock](#) (unsigned int address, array< unsigned int ><sup>^</sup>buffer)  
*Write a GVCP register block.*
- void [ReadGVCPRegisterBlock](#) (unsigned int address, array< unsigned int ><sup>^</sup>buffer)  
*Read a GVCP register block.*

- void [WriteGVCPMemory](#) (unsigned int address, array< unsigned char >^buffer)  
*Write a GVCP memory block.*
- void [ReadGVCPMemory](#) (unsigned int address, array< unsigned char >^buffer)  
*Read a GVCP memory block.*

### GigE property manipulation

These functions deal with GigE properties.

- [GigEProperty](#)^ [GetGigEProperty](#) ([GigEPropertyType](#) propType)  
*Get the specified GigEProperty.*
- void [SetGigEProperty](#) ([GigEProperty](#)^ prop)  
*Set the specified GigEProperty.*
- unsigned int [DiscoverGigEPacketSize](#) ()  
*Discover the largest packet size that works for the network link between the PC and the camera.*

### GigE image settings

These functions deal with GigE image setting.

- bool [QueryGigEImagingMode](#) ([Mode](#) mode)  
*Check if the particular imaging mode is supported by the camera.*
- [Mode](#) [GetGigEImagingMode](#) ()  
*Get the current imaging mode on the camera.*
- void [SetGigEImagingMode](#) ([Mode](#) mode)  
*Set the current imaging mode to the camera.*
- [GigEImageSettingsInfo](#)^ [GetGigEImageSettingsInfo](#) ()  
*Get information about the image settings possible on the camera.*
- [GigEImageSettings](#)^ [GetGigEImageSettings](#) ()  
*Get the current image settings on the camera.*
- void [SetGigEImageSettings](#) ([GigEImageSettings](#)^ settings)  
*Set the image settings specified to the camera.*

### GigE image binning settings

These functions deal with GigE image binning setting.

- void [GetGigEImageBinningSettings](#) (unsigned int% horzBinningValue, unsigned int% vertBinningValue)  
*Get the current binning settings on the camera.*

- void [SetGigEImageBinningSettings](#) (unsigned int horzBinningValue, unsigned int vertBinningValue)

*Set the specified binning values to the camera.*

## GigE image stream configuration

These functions deal with GigE image stream configuration.

- unsigned int [GetNumStreamChannels](#) ()  
*Get the number of stream channels present on the camera.*
- [GigEStreamChannel](#)^ [GetGigEStreamChannelInfo](#) (unsigned int channel)  
*Get the stream channel information for the specified channel.*
- void [SetGigEStreamChannelInfo](#) (unsigned int channel, [GigEStreamChannel](#)^ channelInfo)  
*Set the stream channel information for the specified channel.*

## GigE Configuration

These functions deal with configuring camera.

- [GigEConfig](#)^ [GetGigEConfig](#) ()  
*Get the current configuration on the camera.*
- void [SetGigEConfig](#) ([GigEConfig](#)^ config)  
*Set the configuration specified to the camera.*

### 8.33.1 Detailed Description

The GigECamera object represents a physical Gigabit Ethernet camera.

The object must first be connected to using [Connect\(\)](#) before any other operations can proceed.

Please see [ManagedCameraBase](#) for basic functions that this class inherits from.

### 8.33.2 Constructor & Destructor Documentation

#### 8.33.2.1 ManagedGigECamera ( )

#### 8.33.2.2 ~ManagedGigECamera ( )

#### 8.33.2.3 !ManagedGigECamera ( ) [protected]

### 8.33.3 Member Function Documentation

8.33.3.1 `void Connect ( ManagedPGRGuid^ mgdPGRGuid ) [override, virtual]`

Connects the [ManagedCamera](#) object to the camera specified by the GUID.

#### Parameters

<i>mgdPGR-Guid</i>	The unique identifier for a specific camera on the PC.
--------------------	--

#### See also

[ManagedBusManager::GetCameraFromIndex\(\)](#)  
[ManagedBusManager::GetCameraFromSerialNumber\(\)](#)

Reimplemented from [ManagedCameraBase](#).

8.33.3.2 `unsigned int DiscoverGigEPacketSize ( )`

Discover the largest packet size that works for the network link between the PC and the camera.

This is useful in cases where there may be multiple links between the PC and the camera and there is a possibility of a component not supporting the recommended jumbo frame packet size of 9000.

#### Returns

The maximum packet size supported by the link.

8.33.3.3 `GigEConfig GetGigEConfig ( )`

Get the current configuration on the camera.

#### Returns

Current configuration on camera.

8.33.3.4 `void GetGigEImageBinningSettings ( unsigned int% horzBinningValue, unsigned int% vertBinningValue )`

Get the current binning settings on the camera.

#### Parameters

<i>horzBinning-Value</i>	Current horizontal binning value.
<i>vertBinning-Value</i>	Current vertical binning value.

### 8.33.3.5 **GigImageSettings** GetGigImageSettings ( )

Get the current image settings on the camera.

#### Returns

Current image settings on camera.

### 8.33.3.6 **GigImageSettingsInfo** GetGigImageSettingsInfo ( )

Get information about the image settings possible on the camera.

#### Returns

Image settings information.

### 8.33.3.7 **Mode** GetGigImagingMode ( )

Get the current imaging mode on the camera.

#### Returns

Current imaging mode on the camera.

### 8.33.3.8 **GigEProperty** GetGigEProperty ( **GigEPropertyType** *propType* )

Get the specified [GigEProperty](#).

#### Returns

The GigE property to get.

### 8.33.3.9 **GigEStreamChannel** GetGigEStreamChannelInfo ( unsigned int *channel* )

Get the stream channel information for the specified channel.

#### Parameters

<i>channel</i>	Channel number to use.
----------------	------------------------

#### Returns

Stream channel information for the specified channel.

**8.33.3.10** unsigned int GetNumStreamChannels ( )

Get the number of stream channels present on the camera.

**Returns**

Number of stream channels present.

**8.33.3.11** bool QueryGigEImagingMode ( Mode *mode* )

Check if the particular imaging mode is supported by the camera.

**Parameters**

<i>mode</i>	The mode to check.
-------------	--------------------

**Returns**

Whether the mode is supported.

**8.33.3.12** void ReadGVCPMemory ( unsigned int *address*, array< unsigned char >^ *buffer* )

Read a GVCP memory block.

**Parameters**

<i>address</i>	GVCP address to be read from.
<i>buffer</i>	Array for data to be read into.

**8.33.3.13** unsigned int ReadGVCPRegister ( unsigned int *address* )

Read a GVCP register.

**Parameters**

<i>address</i>	GVCP address to be read from.
----------------	-------------------------------

**Returns**

The value that is read.

**8.33.3.14** void ReadGVCPRegisterBlock ( unsigned int *address*, array< unsigned int >^ *buffer* )

Read a GVCP register block.

## Parameters

<i>address</i>	GVCP address to be read from.
<i>buffer</i>	Array for data to be read into.

8.33.3.15 void SetGigEConfig ( GigEConfig<sup>^</sup> *config* )

Set the configuration specified to the camera.

## Parameters

<i>config</i>	Configuration to set to camera.
---------------	---------------------------------

8.33.3.16 void SetGigEImageBinningSettings ( unsigned int *horzBinningValue*, unsigned int *vertBinningValue* )

Set the specified binning values to the camera.

It is recommended that [GetGigEImageSettingsInfo\(\)](#) be called after this function succeeds to retrieve the new image settings information for the new binning mode.

## Parameters

<i>horz-Binning-Value</i>	Horizontal binning value.
<i>vert-Binning-Value</i>	Vertical binning value.

8.33.3.17 void SetGigEImageSettings ( GigEImageSettings<sup>^</sup> *settings* )

Set the image settings specified to the camera.

## Parameters

<i>settings</i>	Image settings to set to camera.
-----------------	----------------------------------

8.33.3.18 void SetGigEImagingMode ( Mode *mode* )

Set the current imaging mode to the camera.

This should only be done when the camera is not streaming images.



## Parameters

<i>mode</i>	Imaging mode to set to the camera.
-------------	------------------------------------

8.33.3.19 void SetGigEProperty ( GigEProperty<sup>^</sup> *prop* )

Set the specified [GigEProperty](#).

The GigEPropertyType field must be set in order for this function to succeed.

## Parameters

<i>prop</i>	The GigE property to set.
-------------	---------------------------

8.33.3.20 void SetGigEStreamChannelInfo ( unsigned int *channel*, GigEStreamChannel<sup>^</sup> *channelInfo* )

Set the stream channel information for the specified channel.

## Parameters

<i>channel</i>	Channel number to use.
<i>channelInfo</i>	Stream channel information to use for the specified channel.

8.33.3.21 void WriteGVCPMemory ( unsigned int *address*, array< unsigned char ><sup>^</sup> *buffer* )

Write a GVCP memory block.

## Parameters

<i>address</i>	GVCP address to be write to.
<i>buffer</i>	Array containing data to be written.

8.33.3.22 void WriteGVCPRegister ( unsigned int *address*, unsigned int *value* )

Write a GVCP register.

## Parameters

<i>address</i>	GVCP address to be written to.
<i>value</i>	The value to be written.

8.33.3.23 void WriteGVCPRegister ( unsigned int *address*, unsigned int *value*, bool *broadcast* )

Write a GVCP register.

## Parameters

<i>address</i>	GVCP address to be written to.
<i>value</i>	The value to be written.
<i>broadcast</i>	Whether the action should be broadcast.

8.33.3.24 void WriteGVCPRegisterBlock ( unsigned int *address*, array< unsigned int >^ *buffer* )

Write a GVCP register block.

## Parameters

<i>address</i>	GVCP address to be write to.
<i>buffer</i>	Array containing data to be written.

## 8.34 ManagedImage Class Reference

The ManagedImage class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk.

### Public Member Functions

- [ManagedImage](#) ()
- [ManagedImage](#) (unsigned int rows, unsigned int cols, unsigned int stride, unsigned char \*pData, unsigned int dataSize, [PixelFormat](#) format)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, unsigned int stride, unsigned char \*pData, unsigned int dataSize, unsigned int receivedActualSize, - [PixelFormat](#) format)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, unsigned int stride, unsigned char \*pData, unsigned int dataSize, [PixelFormat](#) format, [BayerTileFormat](#) bayerFormat)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, unsigned int stride, unsigned char \*pData, unsigned int dataSize, unsigned int receivedActualSize, - [PixelFormat](#) format, [BayerTileFormat](#) bayerFormat)
- [ManagedImage](#) (unsigned char \*pData, unsigned int dataSize)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, [PixelFormat](#) format)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, [PixelFormat](#) format, [BayerTileFormat](#) bayerFormat)
- [ManagedImage](#) ([ManagedImage](#)^ image)
- [~ManagedImage](#) ()
- void [SetDimensions](#) (unsigned int rows, unsigned int cols, unsigned int stride, [PixelFormat](#) pixelFormat, [BayerTileFormat](#) bayerFormat)  
*Sets the dimensions of the [ManagedImage](#) object.*
- void [GetDimensions](#) (unsigned int \*pRows, unsigned int \*pCols, unsigned int \*pStride, [PixelFormat](#)^ pPixelFormat, [BayerTileFormat](#)^ pBayerFormat)

Get the image dimensions associated with the *ManagedImage* object.

- void **SetData** (unsigned char \*pData, unsigned int dataSize)

Set the data of the *ManagedImage* object.

- void **CalculateStatistics** (*ManagedImageStatistics*^ statistics)

Calculate statistics associated with the image.

- void **Save** (System::String^ fileName)

Save the image to the specified file name.

- void **Save** (System::String^ fileName, *ImageFileFormat* format)

Save the image to the specified file name with the file format specified.

- void **Save** (System::String^ fileName, *PngOption*^ option)

Save the image to the specified file name with the options specified.

- void **Save** (System::String^ fileName, *PpmOption*^ option)

Save the image to the specified file name with the options specified.

- void **Save** (System::String^ fileName, *PgmOption*^ option)

Save the image to the specified file name with the options specified.

- void **Save** (System::String^ fileName, *TiffOption*^ option)

Save the image to the specified file name with the options specified.

- void **Save** (System::String^ fileName, *JpegOption*^ option)

Save the image to the specified file name with the options specified.

- void **Save** (System::String^ fileName, *Jpg2Option*^ option)

Save the image to the specified file name with the options specified.

- void **Save** (System::String^ fileName, *BMPOption*^ option)

Save the image to the specified file name with the options specified.

- void **Convert** (*ManagedImage*^ destImage)

Converts the current image buffer and stores the result in the specified image.

- void **Convert** (*PixelFormat* format, *ManagedImage*^ destImage)

Converts the current image buffer to the specified output format and stores the result in the specified image.

- void **ReleaseBuffer** ()

Release the buffer associated with the *ManagedImage*.

- void \* **GetRawNativeImagePointer** ()

### Static Public Member Functions

- static unsigned int **DetermineBitsPerPixel** (*PixelFormat* format)

Calculate the bits per pixel for the specified pixel format.

### Protected Member Functions

- *!ManagedImage* ()

## Package Functions

- [ManagedImage](#) (FlyCapture2::Image &image)
- bool [IsNativeImageValid](#) ()
- FlyCapture2::Image \* [GetNativeImage](#) ()

## Properties

- static [ColorProcessingAlgorithm](#) [defaultColorProcessingAlgorithm](#) [get, set]  
*The default color processing algorithm to be used.*
- static [PixelFormat](#) [defaultOutputPixelFormat](#) [get, set]  
*The default output pixel format to be used.*
- [ColorProcessingAlgorithm](#) [colorProcessingAlgorithm](#) [get, set]  
*Color processing algorithm to be used.*
- [PixelFormat](#) [pixelFormat](#) [get]  
*Pixel format of the image.*
- [BayerTileFormat](#) [bayerTileFormat](#) [get]  
*Bayer tile format of the image.*
- unsigned int [blockId](#) [get, set]  
*Block id of the image.*
- unsigned int [cols](#) [get]  
*Number of columns in the image.*
- unsigned int [rows](#) [get]  
*Number of rows in the image.*
- unsigned int [stride](#) [get]  
*Number of bytes between rows in the image.*
- unsigned int [bitsPerPixel](#) [get]  
*Number of bits per pixel in the image.*
- unsigned char \* [data](#) [get]  
*Raw pointer to image data.*
- unsigned int [dataSize](#) [get]  
*Size of the buffer associated with the image, in bytes.*
- unsigned int [receivedDataSize](#) [get]  
*Get the size of the compressed data, in bytes.*
- [ImageMetadata](#)<sup>^</sup> [imageMetadata](#) [get]  
*Get the metadata associated with the image.*
- [TimeStamp](#)<sup>^</sup> [timeStamp](#) [get]  
*Get the timestamp data associated with the image.*
- System::Drawing::Bitmap<sup>^</sup> [bitmap](#) [get]  
*Get the internal bitmap representation associated with the image.*

### 8.34.1 Detailed Description

The ManagedImage class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk.

Operations on Image objects are not guaranteed to be thread safe. It is recommended that operations on Image objects be protected by thread synchronization constructs such as mutexes.

### 8.34.2 Constructor & Destructor Documentation

8.34.2.1 **ManagedImage** ( )

8.34.2.2 **ManagedImage** ( unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, unsigned char \* *pData*, unsigned int *dataSize*, PixelFormat *format* )

8.34.2.3 **ManagedImage** ( unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, unsigned char \* *pData*, unsigned int *dataSize*, unsigned int *receivedActualSize*, PixelFormat *format* )

8.34.2.4 **ManagedImage** ( unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, unsigned char \* *pData*, unsigned int *dataSize*, PixelFormat *format*, BayerTileFormat *bayerFormat* )

8.34.2.5 **ManagedImage** ( unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, unsigned char \* *pData*, unsigned int *dataSize*, unsigned int *receivedActualSize*, PixelFormat *format*, BayerTileFormat *bayerFormat* )

8.34.2.6 **ManagedImage** ( unsigned char \* *pData*, unsigned int *dataSize* )

8.34.2.7 **ManagedImage** ( unsigned int *rows*, unsigned int *cols*, PixelFormat *format* )

8.34.2.8 **ManagedImage** ( unsigned int *rows*, unsigned int *cols*, PixelFormat *format*, BayerTileFormat *bayerFormat* )

8.34.2.9 **ManagedImage** ( ManagedImage<sup>^</sup> *image* )

8.34.2.10 **~ManagedImage** ( )

8.34.2.11 **ManagedImage** ( FlyCapture2::Image & *image* ) [package]

8.34.2.12 **!ManagedImage** ( ) [protected]

### 8.34.3 Member Function Documentation

8.34.3.1 **void CalculateStatistics** ( ManagedImageStatistics<sup>^</sup> *statistics* )

Calculate statistics associated with the image.

In order to collect statistics for a particular channel, the enabled flag for the channel must be set to true. Statistics can only be collected for images in Mono8, Mono16, RGB, RGBU, BGR and BGRU.

#### Parameters

<i>statistics</i>	The <a href="#">ManagedImageStatistics</a> object to hold the statistics.
-------------------	---

#### 8.34.3.2 void Convert ( ManagedImage^ destImage )

Converts the current image buffer and stores the result in the specified image.

The destination image does not need to be configured in any way before the call is made.

#### Parameters

<i>destImage</i>	Destination image.
------------------	--------------------

#### 8.34.3.3 void Convert ( PixelFormat format, ManagedImage^ destImage )

Converts the current image buffer to the specified output format and stores the result in the specified image.

The destination image does not need to be configured in any way before the call is made.

#### Parameters

<i>format</i>	Output format of the converted image.
<i>destImage</i>	Destination image.

#### 8.34.3.4 unsigned int DetermineBitsPerPixel ( PixelFormat format ) [static]

Calculate the bits per pixel for the specified pixel format.

#### Parameters

<i>format</i>	The pixel format.
---------------	-------------------

## Returns

The bits per pixel.

8.34.3.5 void GetDimensions ( unsigned int \* *pRows*, unsigned int \* *pCols*, unsigned int \* *pStride*, PixelFormat^ *pPixelFormat*, BayerTileFormat^ *pBayerFormat* )

Get the image dimensions associated with the [ManagedImage](#) object.

## Parameters

<i>pRows</i>	Number of rows.
<i>pCols</i>	Number of columns.
<i>pStride</i>	The stride.
<i>pPixelFormat</i>	Pixel format.
<i>pBayerFormat</i>	Bayer tile format.

8.34.3.6 FlyCapture2::Image \* GetNativeImage ( ) [package]

8.34.3.7 void \* GetRawNativeImagePointer ( )

8.34.3.8 bool IsNativeImageValid ( ) [package]

8.34.3.9 void ReleaseBuffer ( )

Release the buffer associated with the [ManagedImage](#).

If no buffer is associated, the function does nothing.

8.34.3.10 void Save ( System::String^ *fileName* )

Save the image to the specified file name.

## Parameters

<i>fileName</i>	Filename to save image with.
-----------------	------------------------------

8.34.3.11 void Save ( System::String^ *fileName*, ImageFileFormat *format* )

Save the image to the specified file name with the file format specified.

## Parameters

<i>fileName</i>	Filename to save image with.
<i>format</i>	File format to save in.

**8.34.3.12 void Save ( System::String^ fileName, PngOption^ option )**

Save the image to the specified file name with the options specified.

**Parameters**

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

**8.34.3.13 void Save ( System::String^ fileName, PpmOption^ option )**

Save the image to the specified file name with the options specified.

**Parameters**

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

**8.34.3.14 void Save ( System::String^ fileName, PgmOption^ option )**

Save the image to the specified file name with the options specified.

**Parameters**

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

**8.34.3.15 void Save ( System::String^ fileName, TiffOption^ option )**

Save the image to the specified file name with the options specified.

**Parameters**

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

**8.34.3.16 void Save ( System::String^ fileName, JpegOption^ option )**

Save the image to the specified file name with the options specified.

**Parameters**

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.



8.34.3.17 void Save ( System::String^ *fileName*, Jpg2Option^ *option* )

Save the image to the specified file name with the options specified.

Parameters

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

8.34.3.18 void Save ( System::String^ *fileName*, BMPOption^ *option* )

Save the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

8.34.3.19 void SetData ( unsigned char \* *pData*, unsigned int *dataSize* )

Set the data of the [ManagedImage](#) object.

Ownership of the image buffer is not transferred to the [ManagedImage](#) object. It is the user's responsibility to delete the buffer when it is no longer in use.

Parameters

<i>pData</i>	Pointer to the image buffer.
<i>dataSize</i>	Size of the image buffer.

8.34.3.20 void SetDimensions ( unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, PixelFormat *pixelFormat*, BayerTileFormat *bayerFormat* )

Sets the dimensions of the [ManagedImage](#) object.

Parameters

<i>rows</i>	Number of rows to set.
<i>cols</i>	Number of cols to set.
<i>stride</i>	Stride to set.
<i>pixelFormat</i>	Pixel format to set.
<i>bayerFormat</i>	Bayer tile format to set.

## 8.34.4 Property Documentation

**8.34.4.1 BayerTileFormat bayerTileFormat** [get]

Bayer tile format of the image.

**8.34.4.2 System::Drawing::Bitmap^ bitmap** [get]

Get the internal bitmap representation associated with the image.

**Returns**

A System::Drawing::Bitmap containing the image data.

**8.34.4.3 unsigned int bitsPerPixel** [get]

Number of bits per pixel in the image.

**8.34.4.4 unsigned int blockId** [get, set]

Block id of the image.

**8.34.4.5 ColorProcessingAlgorithm colorProcessingAlgorithm** [get, set]

Color processing algorithm to be used.

**8.34.4.6 unsigned int cols** [get]

Number of columns in the image.

**8.34.4.7 unsigned char\* data** [get]

Raw pointer to image data.

**8.34.4.8 unsigned int dataSize** [get]

Size of the buffer associated with the image, in bytes.

**8.34.4.9 ColorProcessingAlgorithm defaultColorProcessingAlgorithm** [static, get, set]

The default color processing algorithm to be used.

**8.34.4.10 PixelFormat defaultOutputPixelFormat** [static, get, set]

The default output pixel format to be used.

**8.34.4.11 ImageMetadata^ imageMetadata** [get]

Get the metadata associated with the image.

This includes embedded image information.

**Returns**

Metadata associated with the image.

**8.34.4.12 PixelFormat pixelFormat** [get]

Pixel format of the image.

**8.34.4.13 unsigned int receivedDataSize** [get]

Get the size of the compressed data, in bytes.

A compressed image will have a maximum size equal to GetDataSize(), but may actually contain less data, depending on the compression level. For uncompressed images, a value smaller than the data size may indicate lost data.

**8.34.4.14 unsigned int rows** [get]

Number of rows in the image.

**8.34.4.15 unsigned int stride** [get]

Number of bytes between rows in the image.

**8.34.4.16 TimeStamp^ timeStamp** [get]

Get the timestamp data associated with the image.

**Returns**

Timestamp data associated with the image.

## 8.35 ManagedImageStatistics Class Reference

### Public Member Functions

- [ManagedImageStatistics](#) ()
- [~ManagedImageStatistics](#) ()
- void [EnableAll](#) ()
- void [DisableAll](#) ()
- void [EnableGreyOnly](#) ()
- void [EnableRGBOnly](#) ()
- void [EnableHSLOnly](#) ()
- bool [GetChannelStatus](#) ([StatisticsChannel](#) channel)
- void [SetChannelStatus](#) ([StatisticsChannel](#) channel, bool enabled)
- void [GetRange](#) ([StatisticsChannel](#) channel, unsigned int% min, unsigned int% max)
- void [GetPixelValueRange](#) ([StatisticsChannel](#) channel, unsigned int% pixelValueMin, unsigned int% pixelValueMax)
- void [GetNumPixelValues](#) ([StatisticsChannel](#) channel, unsigned int% numPixelValues)
- void [GetMean](#) ([StatisticsChannel](#) channel, float% mean)
- void [GetHistogram](#) ([StatisticsChannel](#) channel, array< int >^ histogram)
- void [GetStatistics](#) ([StatisticsChannel](#) channel, unsigned int% rangeMin, unsigned int% rangeMax, unsigned int% pixelValueMin, unsigned int% pixelValueMax, unsigned int% numPixelValues, float% mean, array< int >^ histogram)

### Package Functions

- [FlyCapture2::ImageStatistics \\* GetNativeImageStatistics](#) ()

### 8.35.1 Constructor & Destructor Documentation

#### 8.35.1.1 [ManagedImageStatistics](#) ( )

#### 8.35.1.2 [~ManagedImageStatistics](#) ( )

### 8.35.2 Member Function Documentation

#### 8.35.2.1 void [DisableAll](#) ( )

#### 8.35.2.2 void [EnableAll](#) ( )

#### 8.35.2.3 void [EnableGreyOnly](#) ( )

#### 8.35.2.4 void [EnableHSLOnly](#) ( )

- 8.35.2.5 void EnableRGBOnly ( )
- 8.35.2.6 bool GetChannelStatus ( StatisticsChannel channel )
- 8.35.2.7 void GetHistogram ( StatisticsChannel channel, array< int >^ histogram )
- 8.35.2.8 void GetMean ( StatisticsChannel channel, float% mean )
- 8.35.2.9 FlyCapture2::ImageStatistics \* GetNativeImageStatistics ( ) [package]
- 8.35.2.10 void GetNumPixelValues ( StatisticsChannel channel, unsigned int% numPixelValues )
- 8.35.2.11 void GetPixelValueRange ( StatisticsChannel channel, unsigned int% pixelValueMin, unsigned int% pixelValueMax )
- 8.35.2.12 void GetRange ( StatisticsChannel channel, unsigned int% min, unsigned int% max )
- 8.35.2.13 void GetStatistics ( StatisticsChannel channel, unsigned int% rangeMin, unsigned int% rangeMax, unsigned int% pixelValueMin, unsigned int% pixelValueMax, unsigned int% numPixelValues, float% mean, array< int >^ histogram )
- 8.35.2.14 void SetChannelStatus ( StatisticsChannel channel, bool enabled )

## 8.36 ManagedPGRGuid Class Reference

Managed version of a PGRGuid.

### Public Member Functions

- [ManagedPGRGuid](#) ()  
*Constructor.*
- [ManagedPGRGuid](#) ([ManagedPGRGuid](#)^ managedGuid)  
*Copy constructor.*
- [ManagedPGRGuid](#) ([ManagedPGRGuid](#)% managedGuid)  
*Copy constructor.*
- [ManagedPGRGuid](#)% operator= ([ManagedPGRGuid](#)% managedGuid)  
*Assignment operator.*
- virtual bool [Equals](#) (Object^ obj) override
- virtual int [GetHashCode](#) () override

### Static Public Member Functions

- static bool operator== ([ManagedPGRGuid](#)% left, [ManagedPGRGuid](#)% right)

*Equality operator.*

- static bool `operator!=` (ManagedPGRGuid% left, ManagedPGRGuid% right)

*Inequality operator.*

## Public Attributes

- unsigned int `value0`
- unsigned int `value1`
- unsigned int `value2`
- unsigned int `value3`

### 8.36.1 Detailed Description

Managed version of a PGRGuid.

It is used to uniquely identify a camera.

### 8.36.2 Constructor & Destructor Documentation

#### 8.36.2.1 ManagedPGRGuid ( ) [inline]

Constructor.

#### 8.36.2.2 ManagedPGRGuid ( ManagedPGRGuid^ managedGuid ) [inline]

Copy constructor.

#### 8.36.2.3 ManagedPGRGuid ( ManagedPGRGuid% managedGuid ) [inline]

Copy constructor.

### 8.36.3 Member Function Documentation

#### 8.36.3.1 virtual bool Equals ( Object^ obj ) [inline, override, virtual]

#### 8.36.3.2 virtual int GetHashCode ( ) [inline, override, virtual]

#### 8.36.3.3 static bool operator!= ( ManagedPGRGuid% left, ManagedPGRGuid% right ) [inline, static]

Inequality operator.

8.36.3.4 **ManagedPGRGuid % operator= ( ManagedPGRGuid% *managedGuid* )**  
`[inline]`

Assignment operator.

8.36.3.5 **static bool operator== ( ManagedPGRGuid% *left*, ManagedPGRGuid% *right* )**  
`[inline, static]`

Equality operator.

## 8.36.4 Member Data Documentation

8.36.4.1 unsigned int `value0`

8.36.4.2 unsigned int `value1`

8.36.4.3 unsigned int `value2`

8.36.4.4 unsigned int `value3`

## 8.37 ManagedTopologyNode Class Reference

The [ManagedTopologyNode](#) class contains topology information that can be used to generate a tree structure of all cameras and devices connected to a computer.

### Public Types

- enum [PortType](#) { [NotConnected](#) = 1, [ConnectedToParent](#), [ConnectedToChild](#) }
- Possible states of a port on a node.*
- enum [NodeType](#) { [Computer](#), [Bus](#), [Camera](#), [Node](#) }
- Type of node.*

### Public Member Functions

- virtual [~ManagedTopologyNode](#) ()
- [ManagedTopologyNode](#) ([ManagedTopologyNode](#)^ other)
- [ManagedTopologyNode](#) ([ManagedTopologyNode](#)% other)
- [ManagedPGRGuid](#)^ [GetGuid](#) ()
- Get the PGRGuid associated with the node.*
- int [GetDeviceId](#) ()
- Get the device ID associated with the node.*
- [NodeType](#) [GetNodeType](#) ()

*Get the node type associated with the node.*

- [InterfaceType](#) [GetInterfaceType](#) ()

*Get the interface type associated with the node.*

- unsigned int [GetNumChildren](#) ()

*Get the number of child nodes.*

- [ManagedTopologyNode](#)^ [GetChild](#) (unsigned int position)

*Get child node located at the specified position.*

- unsigned int [GetNumPorts](#) ()

*Get the number of ports.*

- [PortType](#) [GetPortType](#) (unsigned int position)

*Get type of port located at the specified position.*

## Package Functions

- [ManagedTopologyNode](#) (FlyCapture2::TopologyNode \*pNode)

## Static Package Functions

- static [ManagedTopologyNode::PortType](#) [TranslatePortType](#) (FlyCapture2::TopologyNode::PortType portType)
- static FlyCapture2::TopologyNode::PortType [TranslatePortType](#) ([ManagedTopologyNode::PortType](#) portType)
- static [ManagedTopologyNode::NodeType](#) [TranslateNodeType](#) (FlyCapture2::TopologyNode::NodeType portType)
- static FlyCapture2::TopologyNode::NodeType [TranslateNodeType](#) ([ManagedTopologyNode::NodeType](#) portType)

### 8.37.1 Detailed Description

The [ManagedTopologyNode](#) class contains topology information that can be used to generate a tree structure of all cameras and devices connected to a computer.

### 8.37.2 Member Enumeration Documentation

#### 8.37.2.1 enum NodeType

Type of node.

Enumerator:

***Computer***

***Bus***

***Camera***

***Node***



## 8.37.2.2 enum PortType

Possible states of a port on a node.

Enumerator:

***NotConnected***  
***ConnectedToParent***  
***ConnectedToChild***

## 8.37.3 Constructor &amp; Destructor Documentation

8.37.3.1 virtual ~ManagedTopologyNode ( ) [inline, virtual]

8.37.3.2 ManagedTopologyNode ( ManagedTopologyNode<sup>^</sup> other )

8.37.3.3 ManagedTopologyNode ( ManagedTopologyNode% other )

8.37.3.4 ManagedTopologyNode ( FlyCapture2::TopologyNode \* pNode ) [package]

## 8.37.4 Member Function Documentation

8.37.4.1 ManagedTopologyNode GetChild ( unsigned int position )

Get child node located at the specified position.

Parameters

<i>position</i>	Position of the node.
-----------------	-----------------------

Returns

[ManagedTopologyNode](#) at the specified position.

8.37.4.2 int GetDeviceId ( )

Get the device ID associated with the node.

Returns

Device ID of the node.

8.37.4.3 ManagedPGRGuid GetGuid ( )

Get the PGRGuid associated with the node.

**Returns**

PGRGuid of the node.

**8.37.4.4 InterfaceType GetInterfaceType ( )**

Get the interface type associated with the node.

**Returns**

Interface type of the node.

**8.37.4.5 ManagedTopologyNode::NodeType GetNodeType ( )**

Get the node type associated with the node.

**Returns**

Node type of the node.

**8.37.4.6 unsigned int GetNumChildren ( )**

Get the number of child nodes.

**Returns**

Number of child nodes.

**8.37.4.7 unsigned int GetNumPorts ( )**

Get the number of ports.

**Returns**

Number of ports.

**8.37.4.8 ManagedTopologyNode::PortType GetPortType ( unsigned int *position* )**

Get type of port located at the specified position.

**Parameters**

<i>position</i>	Position of the port.
-----------------	-----------------------

## Returns

PortType at the specified position.

- 8.37.4.9 **ManagedTopologyNode::NodeType** TranslateNodeType (   
 FlyCapture2::TopologyNode::NodeType *portType* ) [static, package]
- 8.37.4.10 **FlyCapture2::TopologyNode::NodeType** TranslateNodeType (   
 **ManagedTopologyNode::NodeType** *portType* ) [static, package]
- 8.37.4.11 **ManagedTopologyNode::PortType** TranslatePortType (   
 FlyCapture2::TopologyNode::PortType *portType* ) [static, package]
- 8.37.4.12 **FlyCapture2::TopologyNode::PortType** TranslatePortType (   
 **ManagedTopologyNode::PortType** *portType* ) [static, package]

## 8.38 ManagedUtilities Class Reference

### Static Public Member Functions

- static void [CheckDriver](#) ([ManagedPGRGuid](#)^ guid)
- static System::String^ [GetDriverDeviceName](#) ([ManagedPGRGuid](#)^ mgdPGR-Guid)
- static void [LaunchBrowser](#) (System::String^ address)
- static void [LaunchHelp](#) (System::String^ fileName)
- static void [LaunchCommand](#) (System::String^ command)
- static void [LaunchCommandAsync](#) (System::String^ command, Async-CommandCallback^ hCallbackDelegate)

### Static Package Functions

- static void [OnNativeCallback](#) (FlyCapture2::Error retError, void \*pUserData)

### Static Package Attributes

- static AsyncCommandCallback^ [m\\_externalDelegate](#)
- static CommandCallbackDelegate^ [m\\_internalDelegate](#)

### Properties

- static [SystemInfo](#)^ [systemInfo](#) [get]
- static [FC2Version](#)^ [libraryVersion](#) [get]

### 8.38.1 Member Function Documentation

- 8.38.1.1 `void CheckDriver ( ManagedPGRGuid^ guid ) [static]`
- 8.38.1.2 `System::String GetDriverDeviceName ( ManagedPGRGuid^ mgdPGRGuid ) [static]`
- 8.38.1.3 `void LaunchBrowser ( System::String^ address ) [static]`
- 8.38.1.4 `void LaunchCommand ( System::String^ command ) [static]`
- 8.38.1.5 `void LaunchCommandAsync ( System::String^ command, AsyncCommandCallback^ hCallbackDelegate ) [static]`
- 8.38.1.6 `void LaunchHelp ( System::String^ fileName ) [static]`
- 8.38.1.7 `void OnNativeCallback ( FlyCapture2::Error retError, void * pUserData ) [static, package]`

### 8.38.2 Member Data Documentation

- 8.38.2.1 `AsyncCommandCallback^ m_externalDelegate [static, package]`
- 8.38.2.2 `CommandCallbackDelegate^ m_internalDelegate [static, package]`

### 8.38.3 Property Documentation

- 8.38.3.1 `FC2Version^ libraryVersion [static, get]`
- 8.38.3.2 `SystemInfo^ systemInfo [static, get]`

## 8.39 NativeEventStruct Struct Reference

### Public Attributes

- `FlyCapture2::EventOptions * ptr`

### 8.39.1 Member Data Documentation

- 8.39.1.1 `FlyCapture2::EventOptions* ptr`

## 8.40 PgmOption Struct Reference

Options for saving PGM images.

## Public Member Functions

- [PgmOption](#) ()

## Properties

- bool [binaryFile](#)

*Whether to save the PPM as a binary file.*

### 8.40.1 Detailed Description

Options for saving PGM images.

### 8.40.2 Constructor & Destructor Documentation

#### 8.40.2.1 [PgmOption](#) ( ) `[inline]`

### 8.40.3 Property Documentation

#### 8.40.3.1 bool [binaryFile](#)

Whether to save the PPM as a binary file.

## 8.41 PngOption Struct Reference

Options for saving PNG images.

## Public Member Functions

- [PngOption](#) ()

## Properties

- bool [interlaced](#)  
*Whether to save the PNG as interlaced.*
- unsigned int [compressionLevel](#)  
*Compression level (0-9).*

### 8.41.1 Detailed Description

Options for saving PNG images.

## 8.41.2 Constructor & Destructor Documentation

### 8.41.2.1 PngOption ( ) [inline]

## 8.41.3 Property Documentation

### 8.41.3.1 unsigned int compressionLevel

Compression level (0-9).

0 is no compression, 9 is best compression.

### 8.41.3.2 bool interlaced

Whether to save the PNG as interlaced.

## 8.42 PpmOption Struct Reference

Options for saving PPM images.

### Public Member Functions

- [PpmOption \(\)](#)

### Properties

- bool [binaryFile](#)  
*Whether to save the PPM as a binary file.*

### 8.42.1 Detailed Description

Options for saving PPM images.

## 8.42.2 Constructor & Destructor Documentation

### 8.42.2.1 PpmOption ( ) [inline]

## 8.42.3 Property Documentation

### 8.42.3.1 bool binaryFile

Whether to save the PPM as a binary file.

## 8.43 StrobeControl Struct Reference

A camera strobe.

### Properties

- unsigned int [source](#)  
*Source value.*
- bool [onOff](#)  
*Flag controlling on/off.*
- unsigned int [polarity](#)  
*Signal polarity.*
- float [delay](#)  
*Signal delay (in ms).*
- float [duration](#)  
*Signal duration (in ms).*

### 8.43.1 Detailed Description

A camera strobe.

### 8.43.2 Property Documentation

#### 8.43.2.1 float delay

Signal delay (in ms).

#### 8.43.2.2 float duration

Signal duration (in ms).

#### 8.43.2.3 bool onOff

Flag controlling on/off.

#### 8.43.2.4 unsigned int polarity

Signal polarity.

#### 8.43.2.5 unsigned int source

Source value.

## 8.44 StrobelInfo Struct Reference

A camera strobe property.

### Properties

- unsigned int [source](#)  
*Source value.*
- bool [present](#)  
*Presence of strobe.*
- bool [readOutSupported](#)  
*Flag indicating if strobe value can be read out.*
- bool [onOffSupported](#)  
*Flag indicating if on/off is supported.*
- bool [polaritySupported](#)  
*Flag indicating if polarity is supported.*
- float [minValue](#)  
*Minimum value.*
- float [maxValue](#)  
*Maximum value.*

### 8.44.1 Detailed Description

A camera strobe property.

### 8.44.2 Property Documentation

#### 8.44.2.1 float [maxValue](#)

Maximum value.

#### 8.44.2.2 float [minValue](#)

Minimum value.

#### 8.44.2.3 bool [onOffSupported](#)

Flag indicating if on/off is supported.

#### 8.44.2.4 bool [polaritySupported](#)

Flag indicating if polarity is supported.



## 8.44.2.5 bool present

Presence of strobe.

## 8.44.2.6 bool readOutSupported

Flag indicating if strobe value can be read out.

## 8.44.2.7 unsigned int source

Source value.

## 8.45 SystemInfo Struct Reference

Description of the system.

### Properties

- [OSType osType](#)  
*Operating system type as described by OSType.*
- [System::String^ osDescription](#)  
*Detailed description of the operating system.*
- [ByteOrder byteOrder](#)  
*Byte order of the system.*
- unsigned int [systemMemorySize](#)  
*Amount of memory available on the system.*
- [System::String^ cpuDescription](#)  
*Detailed description of the CPU.*
- unsigned int [numCpuCores](#)  
*Number of cores on all CPUs on the system.*
- [System::String^ driverList](#)  
*List of drivers used.*
- [System::String^ libraryList](#)  
*List of libraries used.*
- [System::String^ gpuDescription](#)  
*Detailed description of the GPU.*
- unsigned int [screenWidth](#)  
*Screen resolution width in pixels.*
- unsigned int [screenHeight](#)  
*Screen resolution height in pixels.*

### 8.45.1 Detailed Description

Description of the system.

### 8.45.2 Property Documentation

#### 8.45.2.1 `ByteOrder` `byteOrder`

Byte order of the system.

#### 8.45.2.2 `System::String^` `cpuDescription`

Detailed description of the CPU.

#### 8.45.2.3 `System::String^` `driverList`

List of drivers used.

#### 8.45.2.4 `System::String^` `gpuDescription`

Detailed description of the GPU.

#### 8.45.2.5 `System::String^` `libraryList`

List of libraries used.

#### 8.45.2.6 `unsigned int` `numCpuCores`

Number of cores on all CPUs on the system.

#### 8.45.2.7 `System::String^` `osDescription`

Detailed description of the operating system.

#### 8.45.2.8 `OSType` `osType`

Operating system type as described by `OSType`.

#### 8.45.2.9 `unsigned int` `screenHeight`

Screen resolution height in pixels.

## 8.45.2.10 unsigned int screenWidth

Screen resolution width in pixels.

## 8.45.2.11 unsigned int systemMemorySize

Amount of memory available on the system.

## 8.46 TiffOption Struct Reference

Options for saving TIFF images.

### Public Types

- enum [CompressionMethod](#) { [None](#) = 1, [PackBits](#), [Deflate](#), [AdobeDeflate](#), [CcittFax3](#), [CcittFax4](#), [Lzw](#), [Jpeg](#) }

### Public Member Functions

- [TiffOption](#) ()

### Properties

- [CompressionMethod](#) *compression*  
*Compression method to use for encoding TIFF images.*

### 8.46.1 Detailed Description

Options for saving TIFF images.

### 8.46.2 Member Enumeration Documentation

8.46.2.1 enum [CompressionMethod](#)

Enumerator:

**None** Save without any compression.

**PackBits** Save using PACKBITS compression.

**Deflate** Save using DEFLATE compression (ZLIB compression).

**AdobeDeflate** Save using ADOBE DEFLATE compression.

**CcittFax3** Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.

**CcittFax4** Save using CCITT Group 4 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.

**Lzw** Save using LZW compression.

**Jpeg** Save using JPEG compression. This is only valid for 8-bit greyscale and 24-bit only. Default to LZW for other bit depths.

### 8.46.3 Constructor & Destructor Documentation

#### 8.46.3.1 TiffOption ( ) [inline]

### 8.46.4 Property Documentation

#### 8.46.4.1 CompressionMethod compression

Compression method to use for encoding TIFF images.

## 8.47 TimeStamp Struct Reference

Timestamp information.

### Properties

- long long [seconds](#)  
*Seconds.*
- unsigned int [microSeconds](#)  
*Microseconds.*
- unsigned int [cycleSeconds](#)  
*1394 cycle time seconds.*
- unsigned int [cycleCount](#)  
*1394 cycle time count.*
- unsigned int [cycleOffset](#)  
*1394 cycle time offset.*

#### 8.47.1 Detailed Description

Timestamp information.

#### 8.47.2 Property Documentation

##### 8.47.2.1 unsigned int cycleCount

1394 cycle time count.

## 8.47.2.2 unsigned int cycleOffset

1394 cycle time offset.

## 8.47.2.3 unsigned int cycleSeconds

1394 cycle time seconds.

## 8.47.2.4 unsigned int microSeconds

Microseconds.

## 8.47.2.5 long long seconds

Seconds.

## 8.48 Translate Class Reference

### Static Package Functions

- static [ErrorType](#) [translate](#) (FlyCapture2::ErrorType errorType)
- static FlyCapture2::ErrorType [translate](#) ([ErrorType](#) errorType)
- static [GrabMode](#) [translate](#) (FlyCapture2::GrabMode grabMode)
- static FlyCapture2::GrabMode [translate](#) ([GrabMode](#) grabMode)
- static [BandwidthAllocation](#) [translate](#) (FlyCapture2::BandwidthAllocation bandwidthAllocation)
- static FlyCapture2::BandwidthAllocation [translate](#) ([BandwidthAllocation](#) bandwidthAllocation)
- static [InterfaceType](#) [translate](#) (FlyCapture2::InterfaceType interfaceType)
- static FlyCapture2::InterfaceType [translate](#) ([InterfaceType](#) interfaceType)
- static [DriverType](#) [translate](#) (FlyCapture2::DriverType driverType)
- static FlyCapture2::DriverType [translate](#) ([DriverType](#) driverType)
- static [PropertyType](#) [translate](#) (FlyCapture2::PropertyType propertyType)
- static FlyCapture2::PropertyType [translate](#) ([PropertyType](#) propertyType)
- static [FrameRate](#) [translate](#) (FlyCapture2::FrameRate frmRate)
- static FlyCapture2::FrameRate [translate](#) ([FrameRate](#) frmRate)
- static [VideoMode](#) [translate](#) (FlyCapture2::VideoMode videoMode)
- static FlyCapture2::VideoMode [translate](#) ([VideoMode](#) videoMode)
- static [PixelFormat](#) [translate](#) (FlyCapture2::PixelFormat pixelFormat)
- static FlyCapture2::PixelFormat [translate](#) ([PixelFormat](#) pixelFormat)
- static [BayerTileFormat](#) [translate](#) (FlyCapture2::BayerTileFormat bayerFormat)
- static FlyCapture2::BayerTileFormat [translate](#) ([BayerTileFormat](#) bayerFormat)
- static [Mode](#) [translate](#) (FlyCapture2::Mode mode)
- static FlyCapture2::Mode [translate](#) ([Mode](#) mode)

- static [BusSpeed](#) [translate](#) (FlyCapture2::BusSpeed busSpeed)
- static FlyCapture2::BusSpeed [translate](#) ([BusSpeed](#) busSpeed)
- static [PCleBusSpeed](#) [translate](#) (FlyCapture2::PCleBusSpeed pcieBusSpeed)
- static FlyCapture2::PCleBusSpeed [translate](#) ([PCleBusSpeed](#) pcieBusSpeed)
- static [ColorProcessingAlgorithm](#) [translate](#) (FlyCapture2::ColorProcessingAlgorithm algorithm)
- static FlyCapture2::ColorProcessingAlgorithm [translate](#) ([ColorProcessingAlgorithm](#) algorithm)
- static [ImageFileFormat](#) [translate](#) (FlyCapture2::ImageFileFormat fileFmt)
- static FlyCapture2::ImageFileFormat [translate](#) ([ImageFileFormat](#) fileFmt)
- static [TiffOption::CompressionMethod](#) [translate](#) (FlyCapture2::TiffOption::CompressionMethod method)
- static FlyCapture2::TiffOption::CompressionMethod [translate](#) ([TiffOption::CompressionMethod](#) method)
- static [StatisticsChannel](#) [translate](#) (FlyCapture2::ImageStatistics::StatisticsChannel channel)
- static FlyCapture2::ImageStatistics::StatisticsChannel [translate](#) ([StatisticsChannel](#) channel)
- static [OSType](#) [translate](#) (FlyCapture2::OSType osType)
- static FlyCapture2::OSType [translate](#) ([OSType](#) osType)
- static [ByteOrder](#) [translate](#) (FlyCapture2::ByteOrder byteOrder)
- static FlyCapture2::ByteOrder [translate](#) ([ByteOrder](#) byteOrder)
- static [GigEPropertyType](#) [translate](#) (FlyCapture2::GigEPropertyType propType)
- static FlyCapture2::GigEPropertyType [translate](#) ([GigEPropertyType](#) propType)
- static void [ToMgd](#) (FlyCapture2::FC2Config \*pNative, [FC2Config](#)^ mgd)
- static void [ToNative](#) ([FC2Config](#)^ mgd, FlyCapture2::FC2Config \*pNative)
- static void [ToMgd](#) (FlyCapture2::PropertyInfo \*pNative, [CameraPropertyInfo](#)^ mgd)
- static void [ToNative](#) ([CameraPropertyInfo](#)^ mgd, FlyCapture2::PropertyInfo \*pNative)
- static void [ToMgd](#) (FlyCapture2::Property \*pNative, [CameraProperty](#)^ mgd)
- static void [ToNative](#) ([CameraProperty](#)^ mgd, FlyCapture2::Property \*pNative)
- static void [ToMgd](#) (FlyCapture2::TriggerModelInfo \*pNative, [TriggerModelInfo](#)^ mgd)
- static void [ToNative](#) ([TriggerModelInfo](#)^ mgd, FlyCapture2::TriggerModelInfo \*pNative)
- static void [ToMgd](#) (FlyCapture2::TriggerMode \*pNative, [TriggerMode](#)^ mgd)
- static void [ToNative](#) ([TriggerMode](#)^ mgd, FlyCapture2::TriggerMode \*pNative)
- static void [ToMgd](#) (FlyCapture2::StrobeInfo \*pNative, [StrobeInfo](#)^ mgd)
- static void [ToNative](#) ([StrobeInfo](#)^ mgd, FlyCapture2::StrobeInfo \*pNative)
- static void [ToMgd](#) (FlyCapture2::StrobeControl \*pNative, [StrobeControl](#)^ mgd)
- static void [ToNative](#) ([StrobeControl](#)^ mgd, FlyCapture2::StrobeControl \*pNative)
- static void [ToMgd](#) (FlyCapture2::Format7ImageSettings \*pNative, [Format7ImageSettings](#)^ mgd)
- static void [ToNative](#) ([Format7ImageSettings](#)^ mgd, FlyCapture2::Format7ImageSettings \*pNative)
- static void [ToMgd](#) (FlyCapture2::Format7Info \*pNative, [Format7Info](#)^ mgd)

- static void [ToNative](#) ([Format7Info](#)^ mgd, FlyCapture2::Format7Info \*pNative)
- static void [ToMgd](#) (FlyCapture2::Format7PacketInfo \*pNative, [Format7PacketInfo](#)^ mgd)
- static void [ToNative](#) ([Format7PacketInfo](#)^ mgd, FlyCapture2::Format7PacketInfo \*pNative)
- static void [ToMgd](#) (FlyCapture2::TimeStamp \*pNative, [TimeStamp](#)^ mgd)
- static void [ToNative](#) ([TimeStamp](#)^ mgd, FlyCapture2::TimeStamp \*pNative)
- static void [ToMgd](#) (FlyCapture2::ConfigROM \*pNative, [ConfigROM](#)^ mgd)
- static void [ToNative](#) ([ConfigROM](#)^ mgd, FlyCapture2::ConfigROM \*pNative)
- static void [ToMgd](#) (FlyCapture2::CameraInfo \*pNative, [CameraInfo](#)^ mgd)
- static void [ToNative](#) ([CameraInfo](#)^ mgd, FlyCapture2::CameraInfo \*pNative)
- static void [ToMgd](#) (FlyCapture2::ImageMetadata \*pNative, [ImageMetadata](#)^ mgd)
- static void [ToNative](#) ([ImageMetadata](#)^ mgd, FlyCapture2::ImageMetadata \*pNative)
- static void [ToMgd](#) (FlyCapture2::LUTData \*pNative, [LutData](#)^ mgd)
- static void [ToNative](#) ([LutData](#)^ mgd, FlyCapture2::LUTData \*pNative)
- static void [ToMgd](#) (FlyCapture2::EmbeddedImageInfoProperty \*pNative, [EmbeddedImageInfoProperty](#)^ mgd)
- static void [ToNative](#) ([EmbeddedImageInfoProperty](#)^ mgd, FlyCapture2::EmbeddedImageInfoProperty \*pNative)
- static void [ToMgd](#) (FlyCapture2::EmbeddedImageInfo \*pNative, [EmbeddedImageInfo](#)^ mgd)
- static void [ToNative](#) ([EmbeddedImageInfo](#)^ mgd, FlyCapture2::EmbeddedImageInfo \*pNative)
- static void [ToMgd](#) (FlyCapture2::PNGOption \*pNative, [PngOption](#)^ mgd)
- static void [ToNative](#) ([PngOption](#)^ mgd, FlyCapture2::PNGOption \*pNative)
- static void [ToMgd](#) (FlyCapture2::PPMOption \*pNative, [PpmOption](#)^ mgd)
- static void [ToNative](#) ([PpmOption](#)^ mgd, FlyCapture2::PPMOption \*pNative)
- static void [ToMgd](#) (FlyCapture2::PGMOption \*pNative, [PgmOption](#)^ mgd)
- static void [ToNative](#) ([PgmOption](#)^ mgd, FlyCapture2::PGMOption \*pNative)
- static void [ToMgd](#) (FlyCapture2::TIFFOption \*pNative, [TiffOption](#)^ mgd)
- static void [ToNative](#) ([TiffOption](#)^ mgd, FlyCapture2::TIFFOption \*pNative)
- static void [ToMgd](#) (FlyCapture2::JPEGOption \*pNative, [JpegOption](#)^ mgd)
- static void [ToNative](#) ([JpegOption](#)^ mgd, FlyCapture2::JPEGOption \*pNative)
- static void [ToMgd](#) (FlyCapture2::JPG2Option \*pNative, [Jpg2Option](#)^ mgd)
- static void [ToNative](#) ([Jpg2Option](#)^ mgd, FlyCapture2::JPG2Option \*pNative)
- static void [ToMgd](#) (FlyCapture2::BMPOption \*pNative, [BMPOption](#)^ mgd)
- static void [ToNative](#) ([BMPOption](#)^ mgd, FlyCapture2::BMPOption \*pNative)
- static void [ToMgd](#) (FlyCapture2::SystemInfo \*pNative, [SystemInfo](#)^ mgd)
- static void [ToMgd](#) (FlyCapture2::FC2Version \*pNative, [FC2Version](#)^ mgd)
- static void [ToMgd](#) (FlyCapture2::IPAddress \*pNative, System::Net::IPAddress^ %mgd)
- static void [ToNative](#) (System::Net::IPAddress^ mgd, FlyCapture2::IPAddress \*pNative)
- static void [ToMgd](#) (FlyCapture2::MACAddress \*pNative, System::Net::NetworkInformation::PhysicalAddress^ %mgd)

- static void [ToNative](#) (System::Net::NetworkInformation::PhysicalAddress^ mgd, - FlyCapture2::MACAddress \*pNative)
- static void [ToMgd](#) (FlyCapture2::GigEProperty \*pNative, [GigEProperty](#)^ mgd)
- static void [ToNative](#) ([GigEProperty](#)^ mgd, FlyCapture2::GigEProperty \*pNative)
- static void [ToMgd](#) (FlyCapture2::GigEImageSettingsInfo \*pNative, [GigEImageSettingsInfo](#)^ mgd)
- static void [ToNative](#) ([GigEImageSettingsInfo](#)^ mgd, FlyCapture2::GigEImageSettingsInfo \*pNative)
- static void [ToMgd](#) (FlyCapture2::GigEImageSettings \*pNative, [GigEImageSettings](#)^ mgd)
- static void [ToNative](#) ([GigEImageSettings](#)^ mgd, FlyCapture2::GigEImageSettings \*pNative)
- static void [Translate::ToMgd](#) (FlyCapture2::GigEConfig \*pNative, [GigEConfig](#)^ mgd)
- static void [Translate::ToNative](#) ([GigEConfig](#)^ mgd, FlyCapture2::GigEConfig \*pNative)
- static void [ToMgd](#) (FlyCapture2::GigEStreamChannel \*pNative, [GigEStreamChannel](#)^ mgd)
- static void [ToNative](#) ([GigEStreamChannel](#)^ mgd, FlyCapture2::GigEStreamChannel \*pNative)
- static void [ToMgd](#) (FlyCapture2::CameraStats \*pNative, [CameraStats](#)^ mgd)

### 8.48.1 Member Function Documentation

- 8.48.1.1 void [ToMgd](#) ( FlyCapture2::FC2Config \* *pNative*, [FC2Config](#)^ *mgd* ) [static, package]
- 8.48.1.2 void [ToMgd](#) ( FlyCapture2::PropertyInfo \* *pNative*, [CameraPropertyInfo](#)^ *mgd* ) [static, package]
- 8.48.1.3 void [ToMgd](#) ( FlyCapture2::Property \* *pNative*, [CameraProperty](#)^ *mgd* ) [static, package]
- 8.48.1.4 void [ToMgd](#) ( FlyCapture2::TriggerModelInfo \* *pNative*, [TriggerModelInfo](#)^ *mgd* ) [static, package]
- 8.48.1.5 void [ToMgd](#) ( FlyCapture2::TriggerMode \* *pNative*, [TriggerMode](#)^ *mgd* ) [static, package]
- 8.48.1.6 void [ToMgd](#) ( FlyCapture2::StrobeInfo \* *pNative*, [StrobeInfo](#)^ *mgd* ) [static, package]
- 8.48.1.7 void [ToMgd](#) ( FlyCapture2::StrobeControl \* *pNative*, [StrobeControl](#)^ *mgd* ) [static, package]
- 8.48.1.8 void [ToMgd](#) ( FlyCapture2::Format7ImageSettings \* *pNative*, [Format7ImageSettings](#)^ *mgd* ) [static, package]



- 8.48.1.9 void ToMgd ( FlyCapture2::Format7Info \* *pNative*, Format7Info<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.10 void ToMgd ( FlyCapture2::Format7PacketInfo \* *pNative*, Format7PacketInfo<sup>^</sup> *mgd* ) [static, package]
- 8.48.1.11 void ToMgd ( FlyCapture2::TimeStamp \* *pNative*, TimeStamp<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.12 void ToMgd ( FlyCapture2::ConfigROM \* *pNative*, ConfigROM<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.13 void ToMgd ( FlyCapture2::CameraInfo \* *pNative*, CameraInfo<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.14 void ToMgd ( FlyCapture2::ImageMetadata \* *pNative*, ImageMetadata<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.15 void ToMgd ( FlyCapture2::LUTData \* *pNative*, LutData<sup>^</sup> *mgd* ) [static, package]
- 8.48.1.16 void ToMgd ( FlyCapture2::EmbeddedImageInfoProperty \* *pNative*, EmbeddedImageInfoProperty<sup>^</sup> *mgd* ) [static, package]
- 8.48.1.17 void ToMgd ( FlyCapture2::EmbeddedImageInfo \* *pNative*, EmbeddedImageInfo<sup>^</sup> *mgd* ) [static, package]
- 8.48.1.18 void ToMgd ( FlyCapture2::PNGOption \* *pNative*, PngOption<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.19 void ToMgd ( FlyCapture2::PPMOption \* *pNative*, PpmOption<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.20 void ToMgd ( FlyCapture2::PGMOption \* *pNative*, PgmOption<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.21 void ToMgd ( FlyCapture2::TIFFOption \* *pNative*, TiffOption<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.22 void ToMgd ( FlyCapture2::JPEGOOption \* *pNative*, JpegOption<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.23 void ToMgd ( FlyCapture2::JPG2Option \* *pNative*, Jpg2Option<sup>^</sup> *mgd* )  
[static, package]
- 8.48.1.24 void ToMgd ( FlyCapture2::BMPOption \* *pNative*, BMPOption<sup>^</sup> *mgd* )  
[static, package]

- 8.48.1.25 `void ToMgd ( FlyCapture2::SystemInfo * pNative, SystemInfo^ mgd )`  
[static, package]
- 8.48.1.26 `void ToMgd ( FlyCapture2::FC2Version * pNative, FC2Version^ mgd )`  
[static, package]
- 8.48.1.27 `void ToMgd ( FlyCapture2::IPAddress * pNative, System::Net::IPAddress^ % mgd )`  
[static, package]
- 8.48.1.28 `void ToMgd ( FlyCapture2::MACAddress * pNative, System::Net::NetworkInformation::PhysicalAddress^ % mgd )` [static, package]
- 8.48.1.29 `void ToMgd ( FlyCapture2::GigEProperty * pNative, GigEProperty^ mgd )`  
[static, package]
- 8.48.1.30 `void ToMgd ( FlyCapture2::GigEImageSettingsInfo * pNative, GigEImageSettingsInfo^ mgd )` [static, package]
- 8.48.1.31 `void ToMgd ( FlyCapture2::GigEImageSettings * pNative, GigEImageSettings^ mgd )` [static, package]
- 8.48.1.32 `void ToMgd ( FlyCapture2::GigEStreamChannel * pNative, GigEStreamChannel^ mgd )` [static, package]
- 8.48.1.33 `void ToMgd ( FlyCapture2::CameraStats * pNative, CameraStats^ mgd )`  
[static, package]
- 8.48.1.34 `void ToNative ( FC2Config^ mgd, FlyCapture2::FC2Config * pNative )`  
[static, package]
- 8.48.1.35 `void ToNative ( CameraPropertyInfo^ mgd, FlyCapture2::PropertyInfo * pNative )`  
[static, package]
- 8.48.1.36 `void ToNative ( CameraProperty^ mgd, FlyCapture2::Property * pNative )`  
[static, package]
- 8.48.1.37 `void ToNative ( TriggerModelInfo^ mgd, FlyCapture2::TriggerModelInfo * pNative )`  
[static, package]
- 8.48.1.38 `void ToNative ( TriggerMode^ mgd, FlyCapture2::TriggerMode * pNative )`  
[static, package]
- 8.48.1.39 `void ToNative ( StrobelInfo^ mgd, FlyCapture2::StrobelInfo * pNative )`  
[static, package]
- 8.48.1.40 `void ToNative ( StrobeControl^ mgd, FlyCapture2::StrobeControl * pNative )`  
[static, package]

- 8.48.1.41 `void ToNative ( Format7ImageSettings^ mgd,  
FlyCapture2::Format7ImageSettings * pNative ) [static, package]`
- 8.48.1.42 `void ToNative ( Format7Info^ mgd, FlyCapture2::Format7Info * pNative )  
[static, package]`
- 8.48.1.43 `void ToNative ( Format7PacketInfo^ mgd, FlyCapture2::Format7PacketInfo *  
pNative ) [static, package]`
- 8.48.1.44 `void ToNative ( TimeStamp^ mgd, FlyCapture2::TimeStamp * pNative )  
[static, package]`
- 8.48.1.45 `void ToNative ( ConfigROM^ mgd, FlyCapture2::ConfigROM * pNative )  
[static, package]`
- 8.48.1.46 `void ToNative ( CameraInfo^ mgd, FlyCapture2::CameraInfo * pNative )  
[static, package]`
- 8.48.1.47 `void ToNative ( ImageMetadata^ mgd, FlyCapture2::ImageMetadata * pNative )  
[static, package]`
- 8.48.1.48 `void ToNative ( LutData^ mgd, FlyCapture2::LUTData * pNative ) [static,  
package]`
- 8.48.1.49 `void ToNative ( EmbeddedImageInfoProperty^ mgd,  
FlyCapture2::EmbeddedImageInfoProperty * pNative ) [static, package]`
- 8.48.1.50 `void ToNative ( EmbeddedImageInfo^ mgd, FlyCapture2::EmbeddedImageInfo *  
pNative ) [static, package]`
- 8.48.1.51 `void ToNative ( PngOption^ mgd, FlyCapture2::PNGOption * pNative )  
[static, package]`
- 8.48.1.52 `void ToNative ( PpmOption^ mgd, FlyCapture2::PPMOption * pNative )  
[static, package]`
- 8.48.1.53 `void ToNative ( PgmOption^ mgd, FlyCapture2::PGMOption * pNative )  
[static, package]`
- 8.48.1.54 `void ToNative ( TiffOption^ mgd, FlyCapture2::TIFFOption * pNative )  
[static, package]`
- 8.48.1.55 `void ToNative ( JpegOption^ mgd, FlyCapture2::JPEGOption * pNative )  
[static, package]`
- 8.48.1.56 `void ToNative ( Jpg2Option^ mgd, FlyCapture2::JPG2Option * pNative )  
[static, package]`

- 8.48.1.57 `void ToNative ( BMPOption^ mgd, FlyCapture2::BMPOption * pNative )`  
[static, package]
- 8.48.1.58 `void ToNative ( System::Net::IPAddress^ mgd, FlyCapture2::IPAddress * pNative )`  
[static, package]
- 8.48.1.59 `void ToNative ( System::Net::NetworkInformation::PhysicalAddress^ mgd,  
FlyCapture2::MACAddress * pNative )` [static, package]
- 8.48.1.60 `void ToNative ( GigEProperty^ mgd, FlyCapture2::GigEProperty * pNative )`  
[static, package]
- 8.48.1.61 `void ToNative ( GigEImageSettingsInfo^ mgd,  
FlyCapture2::GigEImageSettingsInfo * pNative )` [static, package]
- 8.48.1.62 `void ToNative ( GigEImageSettings^ mgd, FlyCapture2::GigEImageSettings *  
pNative )` [static, package]
- 8.48.1.63 `void ToNative ( GigEStreamChannel^ mgd, FlyCapture2::GigEStreamChannel *  
pNative )` [static, package]
- 8.48.1.64 `ErrorType translate ( FlyCapture2::ErrorType errorType )` [static,  
package]
- 8.48.1.65 `FlyCapture2::ErrorType translate ( ErrorType errorType )` [static,  
package]
- 8.48.1.66 `GrabMode translate ( FlyCapture2::GrabMode grabMode )` [static,  
package]
- 8.48.1.67 `FlyCapture2::GrabMode translate ( GrabMode grabMode )` [static,  
package]
- 8.48.1.68 `BandwidthAllocation translate ( FlyCapture2::BandwidthAllocation  
bandwidthAllocation )` [static, package]
- 8.48.1.69 `FlyCapture2::BandwidthAllocation translate ( BandwidthAllocation  
bandwidthAllocation )` [static, package]
- 8.48.1.70 `InterfaceType translate ( FlyCapture2::InterfaceType interfaceType )`  
[static, package]
- 8.48.1.71 `FlyCapture2::InterfaceType translate ( InterfaceType interfaceType )`  
[static, package]
- 8.48.1.72 `DriverType translate ( FlyCapture2::DriverType driverType )` [static,  
package]

- 8.48.1.73 **FlyCapture2::DriverType** translate ( **DriverType** *driverType* ) [static, package]
- 8.48.1.74 **PropertyType** translate ( **FlyCapture2::PropertyType** *propertyType* ) [static, package]
- 8.48.1.75 **FlyCapture2::PropertyType** translate ( **PropertyType** *propertyType* ) [static, package]
- 8.48.1.76 **FrameRate** translate ( **FlyCapture2::FrameRate** *frmRate* ) [static, package]
- 8.48.1.77 **FlyCapture2::FrameRate** translate ( **FrameRate** *frmRate* ) [static, package]
- 8.48.1.78 **VideoMode** translate ( **FlyCapture2::VideoMode** *videoMode* ) [static, package]
- 8.48.1.79 **FlyCapture2::VideoMode** translate ( **VideoMode** *videoMode* ) [static, package]
- 8.48.1.80 **PixelFormat** translate ( **FlyCapture2::PixelFormat** *pixelFormat* ) [static, package]
- 8.48.1.81 **FlyCapture2::PixelFormat** translate ( **PixelFormat** *pixelFormat* ) [static, package]
- 8.48.1.82 **BayerTileFormat** translate ( **FlyCapture2::BayerTileFormat** *bayerFormat* ) [static, package]
- 8.48.1.83 **FlyCapture2::BayerTileFormat** translate ( **BayerTileFormat** *bayerFormat* ) [static, package]
- 8.48.1.84 **Mode** translate ( **FlyCapture2::Mode** *mode* ) [static, package]
- 8.48.1.85 **FlyCapture2::Mode** translate ( **Mode** *mode* ) [static, package]
- 8.48.1.86 **BusSpeed** translate ( **FlyCapture2::BusSpeed** *busSpeed* ) [static, package]
- 8.48.1.87 **FlyCapture2::BusSpeed** translate ( **BusSpeed** *busSpeed* ) [static, package]
- 8.48.1.88 **PCleBusSpeed** translate ( **FlyCapture2::PCleBusSpeed** *pcieBusSpeed* ) [static, package]
- 8.48.1.89 **FlyCapture2::PCleBusSpeed** translate ( **PCleBusSpeed** *pcieBusSpeed* ) [static, package]

- 8.48.1.90 **ColorProcessingAlgorithm** translate ( **FlyCapture2::ColorProcessingAlgorithm** *algorithm* ) [static, package]
- 8.48.1.91 **FlyCapture2::ColorProcessingAlgorithm** translate ( **ColorProcessingAlgorithm** *algorithm* ) [static, package]
- 8.48.1.92 **ImageFileFormat** translate ( **FlyCapture2::ImageFileFormat** *fileFmt* ) [static, package]
- 8.48.1.93 **FlyCapture2::ImageFileFormat** translate ( **ImageFileFormat** *fileFmt* ) [static, package]
- 8.48.1.94 **TiffOption::CompressionMethod** translate ( **FlyCapture2::TiffOption::CompressionMethod** *method* ) [static, package]
- 8.48.1.95 **FlyCapture2::TiffOption::CompressionMethod** translate ( **TiffOption::CompressionMethod** *method* ) [static, package]
- 8.48.1.96 **StatisticsChannel** translate ( **FlyCapture2::ImageStatistics::StatisticsChannel** *channel* ) [static, package]
- 8.48.1.97 **FlyCapture2::ImageStatistics::StatisticsChannel** translate ( **StatisticsChannel** *channel* ) [static, package]
- 8.48.1.98 **OSType** translate ( **FlyCapture2::OSType** *osType* ) [static, package]
- 8.48.1.99 **FlyCapture2::OSType** translate ( **OSType** *osType* ) [static, package]
- 8.48.1.100 **ByteOrder** translate ( **FlyCapture2::ByteOrder** *byteOrder* ) [static, package]
- 8.48.1.101 **FlyCapture2::ByteOrder** translate ( **ByteOrder** *byteOrder* ) [static, package]
- 8.48.1.102 **GigEPropertyType** translate ( **FlyCapture2::GigEPropertyType** *propType* ) [static, package]
- 8.48.1.103 **FlyCapture2::GigEPropertyType** translate ( **GigEPropertyType** *propType* ) [static, package]
- 8.48.1.104 static void Translate::ToMgd ( **FlyCapture2::GigEConfig** \* *pNative*, **GigEConfig**^ *mgd* ) [static, package]
- 8.48.1.105 static void Translate::ToNative ( **GigEConfig**^ *mgd*, **FlyCapture2::GigEConfig** \* *pNative* ) [static, package]

## 8.49 TriggerMode Struct Reference

A camera trigger.

### Properties

- bool [onOff](#)  
*Flag controlling on/off.*
- unsigned int [polarity](#)  
*Polarity value.*
- unsigned int [source](#)  
*Source value.*
- unsigned int [mode](#)  
*Mode value.*
- unsigned int [parameter](#)  
*Parameter value.*

### 8.49.1 Detailed Description

A camera trigger.

### 8.49.2 Property Documentation

#### 8.49.2.1 unsigned int mode

Mode value.

#### 8.49.2.2 bool onOff

Flag controlling on/off.

#### 8.49.2.3 unsigned int parameter

Parameter value.

#### 8.49.2.4 unsigned int polarity

Polarity value.

#### 8.49.2.5 unsigned int source

Source value.

## 8.50 TriggerModelInfo Struct Reference

Information about a camera trigger property.

### Properties

- bool [present](#)  
*Presence of trigger mode.*
- bool [readOutSupported](#)  
*Flag indicating if trigger value can be read out.*
- bool [onOffSupported](#)  
*Flag indicating if on/off is supported.*
- bool [polaritySupported](#)  
*Flag indicating if polarity is supported.*
- bool [valueReadable](#)  
*Flag indicating if the value is readable.*
- unsigned int [sourceMask](#)  
*Source mask.*
- bool [softwareTriggerSupported](#)  
*Flag indicating if software trigger is supported.*
- unsigned int [modeMask](#)  
*Mode mask.*

### 8.50.1 Detailed Description

Information about a camera trigger property.

### 8.50.2 Property Documentation

#### 8.50.2.1 unsigned int modeMask

Mode mask.

#### 8.50.2.2 bool onOffSupported

Flag indicating if on/off is supported.

#### 8.50.2.3 bool polaritySupported

Flag indicating if polarity is supported.



**8.50.2.4 bool present**

Presence of trigger mode.

**8.50.2.5 bool readOutSupported**

Flag indicating if trigger value can be read out.

**8.50.2.6 bool softwareTriggerSupported**

Flag indicating if software trigger is supported.

**8.50.2.7 unsigned int sourceMask**

Source mask.

**8.50.2.8 bool valueReadable**

Flag indicating if the value is readable.

# Index

- ~CameraControlDialog
  - FlyCapture2Managed::Gui::CameraControlDialog, [38](#)
- ~CameraSelectionDialog
  - FlyCapture2Managed::Gui::CameraSelectionDialog, [48](#)
- ~FC2Exception
  - FlyCapture2Managed::FC2Exception, [58](#)
- ~ManagedBusManager
  - FlyCapture2Managed::ManagedBusManager, [76](#)
- ~ManagedCamera
  - FlyCapture2Managed::ManagedCamera, [85](#)
- ~ManagedCameraBase
  - FlyCapture2Managed::ManagedCameraBase, [95](#)
- ~ManagedGCCamera
  - FlyCapture2Managed::ManagedGCCamera, [115](#)
- ~ManagedGCPort
  - FlyCapture2Managed::ManagedGCPort, [117](#)
- ~ManagedGigECamera
  - FlyCapture2Managed::ManagedGigECamera, [120](#)
- ~ManagedImage
  - FlyCapture2Managed::ManagedImage, [129](#)
- ~ManagedImageStatistics
  - FlyCapture2Managed::ManagedImageStatistics, [136](#)
- ~ManagedTopologyNode
  - FlyCapture2Managed::ManagedTopologyNode, [141](#)
- AdobeDeflate
  - FlyCapture2Managed::TiffOption, [151](#)
- Any
  - Enumerations, [16](#)
- Arrival
  - Enumerations, [21](#)
- AutoExposure
  - Enumerations, [24](#)
- BGGR
  - Enumerations, [15](#)
- BigEndian
  - Enumerations, [16](#)
- Blue
  - Enumerations, [24](#)
- Bmp
  - Enumerations, [20](#)
- Brightness
  - Enumerations, [24](#)
- BufferFrames
  - Enumerations, [20](#)
- BufferTooSmall
  - Enumerations, [19](#)
- Bus
  - FlyCapture2Managed::ManagedTopologyNode, [140](#)
- BusMasterFailed
  - Enumerations, [18](#)
- BusReset
  - Enumerations, [21](#)
- Camera
  - FlyCapture2Managed::ManagedTopologyNode, [140](#)
- CcittFax3
  - FlyCapture2Managed::TiffOption, [151](#)
- CcittFax4
  - FlyCapture2Managed::TiffOption, [151](#)
- Computer
  - FlyCapture2Managed::ManagedTopologyNode, [140](#)
- ConnectedToChild
  - FlyCapture2Managed::ManagedTopologyNode, [141](#)
- ConnectedToParent

- FlyCapture2Managed::Managed-TopologyNode, [141](#)
- Default
  - Enumerations, [17](#)
- Deflate
  - FlyCapture2Managed::TiffOption, [151](#)
- Directional
  - Enumerations, [17](#)
- DropFrames
  - Enumerations, [20](#)
- EdgeSensing
  - Enumerations, [17](#)
- Enumerations
  - Any, [16](#)
  - Arrival, [21](#)
  - AutoExposure, [24](#)
  - BGGR, [15](#)
  - BigEndian, [16](#)
  - Blue, [24](#)
  - Bmp, [20](#)
  - Brightness, [24](#)
  - BufferFrames, [20](#)
  - BufferTooSmall, [19](#)
  - BusMasterFailed, [18](#)
  - BusReset, [21](#)
  - Default, [17](#)
  - Directional, [17](#)
  - DropFrames, [20](#)
  - EdgeSensing, [17](#)
  - Failed, [18](#)
  - FailedBusMasterConnection, [18](#)
  - FailedGuid, [18](#)
  - Fastest, [16](#)
  - Focus, [24](#)
  - FrameRate, [24](#)
  - FrameRate120, [19](#)
  - FrameRate15, [19](#)
  - FrameRate1\_875, [19](#)
  - FrameRate240, [19](#)
  - FrameRate30, [19](#)
  - FrameRate3\_75, [19](#)
  - FrameRate60, [19](#)
  - FrameRate7\_5, [19](#)
  - FrameRateFormat7, [19](#)
  - FromFileExtension, [20](#)
  - GBRG, [15](#)
  - GRBG, [15](#)
  - Gain, [24](#)
  - Gamma, [24](#)
  - GigE, [21](#)
  - GigE\_10000Base\_T, [16](#)
  - GigE\_1000Base\_T, [16](#)
  - GigE\_100Base\_T, [16](#)
  - GigE\_10Base\_T, [16](#)
  - GigE\_Filter, [17](#)
  - GigE\_Lwf, [17](#)
  - GigE\_None, [17](#)
  - GigE\_Pro, [17](#)
  - Green, [24](#)
  - Grey, [24](#)
  - HQLinear, [17](#)
  - Heartbeat, [19](#)
  - HeartbeatTimeout, [19](#)
  - Hue, [24](#)
  - IPP, [17](#)
  - ieee1394, [21](#)
  - ieee1394\_Cam, [17](#)
  - ieee1394\_Juju, [17](#)
  - ieee1394\_Pro, [17](#)
  - ieee1394\_Raw1394, [17](#)
  - ieee1394\_Video1394, [17](#)
  - lidcFailed, [18](#)
  - ImageConsistencyError, [19](#)
  - ImageConversionFailed, [18](#)
  - ImageLibraryFailure, [19](#)
  - IncompatibleDriver, [19](#)
  - Infinite, [20](#)
  - InitFailed, [18](#)
  - InvalidBuManager, [18](#)
  - InvalidGeneration, [18](#)
  - InvalidMode, [18](#)
  - InvalidPacketSize, [18](#)
  - InvalidParameter, [18](#)
  - InvalidSettings, [18](#)
  - Iris, [24](#)
  - IsochAlreadyStarted, [18](#)
  - IsochBandwidthExceeded, [18](#)
  - IsochFailed, [18](#)
  - IsochNotStarted, [18](#)
  - IsochRetrieveBufferFailed, [18](#)
  - IsochStartFailed, [18](#)
  - IsochStopFailed, [18](#)
  - IsochSyncFailed, [18](#)
  - Jpeg, [20](#)
  - Jpeg2000, [21](#)
  - Lightness, [24](#)
  - LinuxX64, [22](#)
  - LinuxX86, [22](#)
  - LittleEndian, [16](#)

- LowLevelFailure, 18
- LutFailed, 18
- Mac, 22
- MemoryAllocationFailed, 18
- Mode0, 21
- Mode1, 21
- Mode10, 22
- Mode11, 22
- Mode12, 22
- Mode13, 22
- Mode14, 22
- Mode15, 22
- Mode16, 22
- Mode17, 22
- Mode18, 22
- Mode19, 22
- Mode2, 21
- Mode20, 22
- Mode21, 22
- Mode22, 22
- Mode23, 22
- Mode24, 22
- Mode25, 22
- Mode26, 22
- Mode27, 22
- Mode28, 22
- Mode29, 22
- Mode3, 21
- Mode30, 22
- Mode31, 22
- Mode4, 21
- Mode5, 21
- Mode6, 22
- Mode7, 22
- Mode8, 22
- Mode9, 22
- NearestNeighbor, 17
- NoColorProcessing, 17
- None, 15, 20
- NotConnected, 18
- NotFound, 18
- NotImplemented, 18
- NotInFormat7, 18
- NotInitialized, 18
- NotSupported, 18
- NumberOfFrameRates, 19
- NumberOfModes, 22
- NumberOfPixelFormats, 23
- NumberOfStatisticsChannels, 24
- NumberOfVideoModes, 25
- Off, 15
- Ok, 17
- On, 15
- PacketDelay, 19
- PacketSize, 19
- Pan, 24
- Pgm, 20
- PixelFormat411Yuv8, 23
- PixelFormat422Yuv8, 23
- PixelFormat422Yuv8Jpeg, 23
- PixelFormat444Yuv8, 23
- PixelFormatBgr, 23
- PixelFormatBgr16, 23
- PixelFormatBgru, 23
- PixelFormatBgru16, 23
- PixelFormatMono12, 23
- PixelFormatMono16, 23
- PixelFormatMono8, 23
- PixelFormatRaw12, 23
- PixelFormatRaw16, 23
- PixelFormatRaw8, 23
- PixelFormatRgb, 23
- PixelFormatRgb16, 23
- PixelFormatRgb8, 23
- PixelFormatRgba, 23
- PixelFormatSignedMono16, 23
- PixelFormatSignedRgb16, 23
- Png, 21
- Ppm, 20
- PropertyFailed, 18
- PropertyNotPresent, 18
- RGBA, 15
- Raw, 21
- ReadRegisterFailed, 18
- Red, 24
- RegisterFailed, 18
- Removal, 21
- Rigorous, 17
- S100, 16
- S1600, 16
- S200, 16
- S3200, 16
- S400, 16
- S480, 16
- S5000, 16
- S800, 16
- Saturation, 24
- Sharpness, 24
- Shutter, 24
- Speed\_2\_5, 23

- Speed\_5\_0, [23](#)
- StrobeFailed, [18](#)
- Temperature, [24](#)
- Tiff, [21](#)
- Tilt, [24](#)
- Timeout, [18](#)
- TriggerDelay, [24](#)
- TriggerFailed, [18](#)
- TriggerMode, [24](#)
- Undefined, [17](#)
- Unknown, [16](#), [17](#), [21](#), [23](#)
- UnknownOS, [22](#)
- Unspecified, [15](#), [20](#), [24](#)
- Unsupported, [15](#)
- Usb2, [21](#)
- Usb3, [21](#)
- Usb3\_Pro, [17](#)
- Usb\_Cam, [17](#)
- Usb\_None, [17](#)
- VideoMode1024x768Rgb, [25](#)
- VideoMode1024x768Y16, [25](#)
- VideoMode1024x768Y8, [25](#)
- VideoMode1024x768Yuv422, [25](#)
- VideoMode1280x960Rgb, [25](#)
- VideoMode1280x960Y16, [25](#)
- VideoMode1280x960Y8, [25](#)
- VideoMode1280x960Yuv422, [25](#)
- VideoMode1600x1200Rgb, [25](#)
- VideoMode1600x1200Y16, [25](#)
- VideoMode1600x1200Y8, [25](#)
- VideoMode1600x1200Yuv422, [25](#)
- VideoMode160x120Yuv444, [25](#)
- VideoMode320x240Yuv422, [25](#)
- VideoMode640x480Rgb, [25](#)
- VideoMode640x480Y16, [25](#)
- VideoMode640x480Y8, [25](#)
- VideoMode640x480Yuv411, [25](#)
- VideoMode640x480Yuv422, [25](#)
- VideoMode800x600Rgb, [25](#)
- VideoMode800x600Y16, [25](#)
- VideoMode800x600Y8, [25](#)
- VideoMode800x600Yuv422, [25](#)
- VideoModeFormat7, [25](#)
- WeightedDirectional, [17](#)
- WhiteBalance, [24](#)
- WindowsX64, [22](#)
- WindowsX86, [22](#)
- WriteRegisterFailed, [18](#)
- Zoom, [24](#)
- Failed
  - Enumerations, [18](#)
- FailedBusMasterConnection
  - Enumerations, [18](#)
- FailedGuid
  - Enumerations, [18](#)
- Fastest
  - Enumerations, [16](#)
- FlyCapture2Managed::ManagedTopology-Node
  - Bus, [140](#)
  - Camera, [140](#)
  - Computer, [140](#)
  - ConnectedToChild, [141](#)
  - ConnectedToParent, [141](#)
  - Node, [140](#)
  - NotConnected, [141](#)
- FlyCapture2Managed::TiffOption
  - AdobeDeflate, [151](#)
  - CcittFax3, [151](#)
  - CcittFax4, [151](#)
  - Deflate, [151](#)
  - Jpeg, [152](#)
  - Lzw, [152](#)
  - None, [151](#)
  - PackBits, [151](#)
- Focus
  - Enumerations, [24](#)
- FrameRate
  - Enumerations, [24](#)
- FrameRate120
  - Enumerations, [19](#)
- FrameRate15
  - Enumerations, [19](#)
- FrameRate1\_875
  - Enumerations, [19](#)
- FrameRate240
  - Enumerations, [19](#)
- FrameRate30
  - Enumerations, [19](#)
- FrameRate3\_75
  - Enumerations, [19](#)
- FrameRate60
  - Enumerations, [19](#)
- FrameRate7\_5
  - Enumerations, [19](#)
- FrameRateFormat7
  - Enumerations, [19](#)
- FromFileExtension
  - Enumerations, [20](#)
- GBRG

- Enumerations, [15](#)
- GRBG
  - Enumerations, [15](#)
- Gain
  - Enumerations, [24](#)
- Gamma
  - Enumerations, [24](#)
- GigE
  - Enumerations, [21](#)
- GigE\_10000Base\_T
  - Enumerations, [16](#)
- GigE\_1000Base\_T
  - Enumerations, [16](#)
- GigE\_100Base\_T
  - Enumerations, [16](#)
- GigE\_10Base\_T
  - Enumerations, [16](#)
- GigE\_Filter
  - Enumerations, [17](#)
- GigE\_Lwf
  - Enumerations, [17](#)
- GigE\_None
  - Enumerations, [17](#)
- GigE\_Pro
  - Enumerations, [17](#)
- Green
  - Enumerations, [24](#)
- Grey
  - Enumerations, [24](#)
- HQLinear
  - Enumerations, [17](#)
- Heartbeat
  - Enumerations, [19](#)
- HeartbeatTimeout
  - Enumerations, [19](#)
- Hue
  - Enumerations, [24](#)
- IPP
  - Enumerations, [17](#)
- leee1394
  - Enumerations, [21](#)
- leee1394\_Cam
  - Enumerations, [17](#)
- leee1394\_Juju
  - Enumerations, [17](#)
- leee1394\_Pro
  - Enumerations, [17](#)
- leee1394\_Raw1394
  - Enumerations, [17](#)
- leee1394\_Video1394
  - Enumerations, [17](#)
- lircFailed
  - Enumerations, [18](#)
- ImageConsistencyError
  - Enumerations, [19](#)
- ImageConversionFailed
  - Enumerations, [18](#)
- ImageLibraryFailure
  - Enumerations, [19](#)
- IncompatibleDriver
  - Enumerations, [19](#)
- Infinite
  - Enumerations, [20](#)
- InitFailed
  - Enumerations, [18](#)
- InvalidBuManager
  - Enumerations, [18](#)
- InvalidGeneration
  - Enumerations, [18](#)
- InvalidMode
  - Enumerations, [18](#)
- InvalidPacketSize
  - Enumerations, [18](#)
- InvalidParameter
  - Enumerations, [18](#)
- InvalidSettings
  - Enumerations, [18](#)
- Iris
  - Enumerations, [24](#)
- IsochAlreadyStarted
  - Enumerations, [18](#)
- IsochBandwidthExceeded
  - Enumerations, [18](#)
- IsochFailed
  - Enumerations, [18](#)
- IsochNotStarted
  - Enumerations, [18](#)
- IsochRetrieveBufferFailed
  - Enumerations, [18](#)
- IsochStartFailed
  - Enumerations, [18](#)
- IsochStopFailed
  - Enumerations, [18](#)
- IsochSyncFailed
  - Enumerations, [18](#)
- Jpeg
  - Enumerations, [20](#)
  - FlyCapture2Managed::TiffOption, [152](#)
- Jpeg2000

- Enumerations, [21](#)
- Lightness
  - Enumerations, [24](#)
- LinuxX64
  - Enumerations, [22](#)
- LinuxX86
  - Enumerations, [22](#)
- LittleEndian
  - Enumerations, [16](#)
- LowLevelFailure
  - Enumerations, [18](#)
- LutFailed
  - Enumerations, [18](#)
- Lzw
  - FlyCapture2Managed::TiffOption, [152](#)
- Mac
  - Enumerations, [22](#)
- MemoryAllocationFailed
  - Enumerations, [18](#)
- Mode0
  - Enumerations, [21](#)
- Mode1
  - Enumerations, [21](#)
- Mode10
  - Enumerations, [22](#)
- Mode11
  - Enumerations, [22](#)
- Mode12
  - Enumerations, [22](#)
- Mode13
  - Enumerations, [22](#)
- Mode14
  - Enumerations, [22](#)
- Mode15
  - Enumerations, [22](#)
- Mode16
  - Enumerations, [22](#)
- Mode17
  - Enumerations, [22](#)
- Mode18
  - Enumerations, [22](#)
- Mode19
  - Enumerations, [22](#)
- Mode2
  - Enumerations, [21](#)
- Mode20
  - Enumerations, [22](#)
- Mode21
  - Enumerations, [22](#)
- Mode22
  - Enumerations, [22](#)
- Mode23
  - Enumerations, [22](#)
- Mode24
  - Enumerations, [22](#)
- Mode25
  - Enumerations, [22](#)
- Mode26
  - Enumerations, [22](#)
- Mode27
  - Enumerations, [22](#)
- Mode28
  - Enumerations, [22](#)
- Mode29
  - Enumerations, [22](#)
- Mode3
  - Enumerations, [21](#)
- Mode30
  - Enumerations, [22](#)
- Mode31
  - Enumerations, [22](#)
- Mode4
  - Enumerations, [21](#)
- Mode5
  - Enumerations, [21](#)
- Mode6
  - Enumerations, [22](#)
- Mode7
  - Enumerations, [22](#)
- Mode8
  - Enumerations, [22](#)
- Mode9
  - Enumerations, [22](#)
- NearestNeighbor
  - Enumerations, [17](#)
- NoColorProcessing
  - Enumerations, [17](#)
- Node
  - FlyCapture2Managed::Managed-TopologyNode, [140](#)
- None
  - Enumerations, [15](#), [20](#)
  - FlyCapture2Managed::TiffOption, [151](#)
- NotConnected
  - Enumerations, [18](#)
  - FlyCapture2Managed::Managed-TopologyNode, [141](#)
- NotFound

- Enumerations, [18](#)
- NotImplemented
  - Enumerations, [18](#)
- NotInFormat7
  - Enumerations, [18](#)
- NotInitialized
  - Enumerations, [18](#)
- NotSupported
  - Enumerations, [18](#)
- NumberOfFrameRates
  - Enumerations, [19](#)
- NumberOfModes
  - Enumerations, [22](#)
- NumberOfPixelFormats
  - Enumerations, [23](#)
- NumberOfStatisticsChannels
  - Enumerations, [24](#)
- NumberOfVideoModes
  - Enumerations, [25](#)
- Off
  - Enumerations, [15](#)
- Ok
  - Enumerations, [17](#)
- On
  - Enumerations, [15](#)
- PackBits
  - FlyCapture2Managed::TiffOption, [151](#)
- PacketDelay
  - Enumerations, [19](#)
- PacketSize
  - Enumerations, [19](#)
- Pan
  - Enumerations, [24](#)
- Pgm
  - Enumerations, [20](#)
- PixelFormat411Yuv8
  - Enumerations, [23](#)
- PixelFormat422Yuv8
  - Enumerations, [23](#)
- PixelFormat422Yuv8Jpeg
  - Enumerations, [23](#)
- PixelFormat444Yuv8
  - Enumerations, [23](#)
- PixelFormatBgr
  - Enumerations, [23](#)
- PixelFormatBgr16
  - Enumerations, [23](#)
- PixelFormatBgru
  - Enumerations, [23](#)
- PixelFormatBgru16
  - Enumerations, [23](#)
- PixelFormatMono12
  - Enumerations, [23](#)
- PixelFormatMono16
  - Enumerations, [23](#)
- PixelFormatMono8
  - Enumerations, [23](#)
- PixelFormatRaw12
  - Enumerations, [23](#)
- PixelFormatRaw16
  - Enumerations, [23](#)
- PixelFormatRaw8
  - Enumerations, [23](#)
- PixelFormatRgb
  - Enumerations, [23](#)
- PixelFormatRgb16
  - Enumerations, [23](#)
- PixelFormatRgb8
  - Enumerations, [23](#)
- PixelFormatRgba
  - Enumerations, [23](#)
- PixelFormatSignedMono16
  - Enumerations, [23](#)
- PixelFormatSignedRgb16
  - Enumerations, [23](#)
- Png
  - Enumerations, [21](#)
- Ppm
  - Enumerations, [20](#)
- PropertyFailed
  - Enumerations, [18](#)
- PropertyNotPresent
  - Enumerations, [18](#)
- RGGB
  - Enumerations, [15](#)
- Raw
  - Enumerations, [21](#)
- ReadRegisterFailed
  - Enumerations, [18](#)
- Red
  - Enumerations, [24](#)
- RegisterFailed
  - Enumerations, [18](#)
- Removal
  - Enumerations, [21](#)
- Rigorous
  - Enumerations, [17](#)
- S100
  - Enumerations, [16](#)



- S1600
  - Enumerations, [16](#)
- S200
  - Enumerations, [16](#)
- S3200
  - Enumerations, [16](#)
- S400
  - Enumerations, [16](#)
- S480
  - Enumerations, [16](#)
- S5000
  - Enumerations, [16](#)
- S800
  - Enumerations, [16](#)
- Saturation
  - Enumerations, [24](#)
- Sharpness
  - Enumerations, [24](#)
- Shutter
  - Enumerations, [24](#)
- Speed\_2\_5
  - Enumerations, [23](#)
- Speed\_5\_0
  - Enumerations, [23](#)
- StrobeFailed
  - Enumerations, [18](#)
- Temperature
  - Enumerations, [24](#)
- Tiff
  - Enumerations, [21](#)
- Tilt
  - Enumerations, [24](#)
- Timeout
  - Enumerations, [18](#)
- TriggerDelay
  - Enumerations, [24](#)
- TriggerFailed
  - Enumerations, [18](#)
- TriggerMode
  - Enumerations, [24](#)
- Undefined
  - Enumerations, [17](#)
- Unknown
  - Enumerations, [16](#), [17](#), [21](#), [23](#)
- UnknownOS
  - Enumerations, [22](#)
- Unspecified
  - Enumerations, [15](#), [20](#), [24](#)
- Unsupported
  - Enumerations, [15](#)
- Usb2
  - Enumerations, [21](#)
- Usb3
  - Enumerations, [21](#)
- Usb3\_Pro
  - Enumerations, [17](#)
- Usb\_Cam
  - Enumerations, [17](#)
- Usb\_None
  - Enumerations, [17](#)
- VideoMode1024x768Rgb
  - Enumerations, [25](#)
- VideoMode1024x768Y16
  - Enumerations, [25](#)
- VideoMode1024x768Y8
  - Enumerations, [25](#)
- VideoMode1024x768Yuv422
  - Enumerations, [25](#)
- VideoMode1280x960Rgb
  - Enumerations, [25](#)
- VideoMode1280x960Y16
  - Enumerations, [25](#)
- VideoMode1280x960Y8
  - Enumerations, [25](#)
- VideoMode1280x960Yuv422
  - Enumerations, [25](#)
- VideoMode1600x1200Rgb
  - Enumerations, [25](#)
- VideoMode1600x1200Y16
  - Enumerations, [25](#)
- VideoMode1600x1200Y8
  - Enumerations, [25](#)
- VideoMode1600x1200Yuv422
  - Enumerations, [25](#)
- VideoMode160x120Yuv444
  - Enumerations, [25](#)
- VideoMode320x240Yuv422
  - Enumerations, [25](#)
- VideoMode640x480Rgb
  - Enumerations, [25](#)
- VideoMode640x480Y16
  - Enumerations, [25](#)
- VideoMode640x480Y8
  - Enumerations, [25](#)
- VideoMode640x480Yuv411
  - Enumerations, [25](#)
- VideoMode640x480Yuv422
  - Enumerations, [25](#)
- VideoMode800x600Rgb
  - Enumerations, [25](#)

- VideoMode800x600Y16
  - Enumerations, [25](#)
- VideoMode800x600Y8
  - Enumerations, [25](#)
- VideoMode800x600Yuv422
  - Enumerations, [25](#)
- VideoModeFormat7
  - Enumerations, [25](#)
- WeightedDirectional
  - Enumerations, [17](#)
- WhiteBalance
  - Enumerations, [24](#)
- WindowsX64
  - Enumerations, [22](#)
- WindowsX86
  - Enumerations, [22](#)
- WriteRegisterFailed
  - Enumerations, [18](#)
- Zoom
  - Enumerations, [24](#)
- AsyncCommandCallback
  - FlyCapture2Managed, [34](#)
- BMPOption, [37](#)
  - FlyCapture2Managed::BMPOption, [37](#)
- BandwidthAllocation
  - Enumerations, [15](#)
- BayerTileFormat
  - Enumerations, [15](#)
- BusSpeed
  - Enumerations, [15](#)
- ByteOrder
  - Enumerations, [16](#)
- CalculateStatistics
  - FlyCapture2Managed::Managed-Image, [129](#)
- CameraControlDialog, [37](#)
  - FlyCapture2Managed::Gui::Camera-ControlDialog, [38](#)
- CameraInfo, [38](#)
- CameraProperty, [43](#)
  - FlyCapture2Managed::Camera-Property, [44](#)
- CameraPropertyInfo, [45](#)
  - FlyCapture2Managed::Camera-PropertyInfo, [46](#)
- CameraSelectionDialog, [48](#)
  - FlyCapture2Managed::Gui::Camera-SelectionDialog, [48](#)
- CameraStats, [49](#)
  - FlyCapture2Managed::CameraStats, [50](#)
- CauseType
  - FlyCapture2Managed::FC2Exception, [58](#)
- CheckDriver
  - FlyCapture2Managed::Managed-Utilities, [144](#)
- ColorProcessingAlgorithm
  - Enumerations, [16](#)
- CommandCallbackDelegate
  - FlyCapture2Managed, [34](#)
- CompressionMethod
  - FlyCapture2Managed::TiffOption, [151](#)
- ConfigROM, [51](#)
- Connect
  - FlyCapture2Managed::Gui::Camera-ControlDialog, [38](#)
  - FlyCapture2Managed::Managed-Camera, [86](#)
  - FlyCapture2Managed::Managed-CameraBase, [95](#)
  - FlyCapture2Managed::ManagedGC-Camera, [115](#), [116](#)
  - FlyCapture2Managed::ManagedGig-ECamera, [120](#)
- Convert
  - FlyCapture2Managed::Managed-Image, [130](#)
- ConvertToManagedGuid
  - FlyCapture2Managed::Managed-BusManager, [76](#)
- ConvertToNativeGuid
  - FlyCapture2Managed::Managed-BusManager, [76](#)
- DeregisterAllEvents
  - FlyCapture2Managed::Managed-CameraBase, [95](#)
- DeregisterEvent
  - FlyCapture2Managed::Managed-CameraBase, [95](#)
- DetermineBitsPerPixel
  - FlyCapture2Managed::Managed-Image, [130](#)
- DisableAll
  - FlyCapture2Managed::Managed-ImageStatistics, [136](#)
- Disconnect

- FlyCapture2Managed::Gui::Camera-  
ControlDialog, [38](#)
- FlyCapture2Managed::Managed-  
CameraBase, [95](#)
- FlyCapture2Managed::ManagedGC-  
Camera, [116](#)
- DiscoverGigECameras
  - FlyCapture2Managed::Managed-  
BusManager, [77](#)
- DiscoverGigEPacketSize
  - FlyCapture2Managed::ManagedGig-  
ECamera, [121](#)
- DriverType
  - Enumerations, [17](#)
- EmbeddedImageInfo, [53](#)
  - FlyCapture2Managed::Embedded-  
ImageInfo, [53](#)
- EmbeddedImageInfoProperty, [54](#)
- EnableAll
  - FlyCapture2Managed::Managed-  
ImageStatistics, [136](#)
- EnableGreyOnly
  - FlyCapture2Managed::Managed-  
ImageStatistics, [136](#)
- EnableHSLOnly
  - FlyCapture2Managed::Managed-  
ImageStatistics, [136](#)
- EnableLUT
  - FlyCapture2Managed::Managed-  
CameraBase, [96](#)
- EnableRGBOnly
  - FlyCapture2Managed::Managed-  
ImageStatistics, [136](#)
- EnumCallback
  - FlyCapture2Managed, [34](#)
- Enumerations, [13](#)
  - BandwidthAllocation, [15](#)
  - BayerTileFormat, [15](#)
  - BusSpeed, [15](#)
  - ByteOrder, [16](#)
  - ColorProcessingAlgorithm, [16](#)
  - DriverType, [17](#)
  - ErrorType, [17](#)
  - FrameRate, [19](#)
  - GigEPropertyType, [19](#)
  - GrabMode, [19](#)
  - GrabTimeout, [20](#)
  - ImageFileFormat, [20](#)
  - InterfaceType, [21](#)
  - ManagedCallbackType, [21](#)
  - Mode, [21](#)
  - OSType, [22](#)
  - PCleBusSpeed, [22](#)
  - PixelFormat, [23](#)
  - PropertyType, [23](#)
  - StatisticsChannel, [24](#)
  - VideoMode, [24](#)
- Equals
  - FlyCapture2Managed::ManagedPG-  
RGuid, [138](#)
- ErrorType
  - Enumerations, [17](#)
- EventCallbackFcn
  - FlyCapture2Managed::Managed-  
EventOptions, [114](#)
- EventID
  - FlyCapture2Managed::Managed-  
EventCallbackData, [113](#)
- EventName
  - FlyCapture2Managed::Managed-  
EventCallbackData, [113](#)
  - FlyCapture2Managed::Managed-  
EventOptions, [114](#)
- EventTimestamp
  - FlyCapture2Managed::Managed-  
EventCallbackData, [113](#)
- FC2Config, [54](#)
  - FlyCapture2Managed::FC2Config,  
[55](#)
- FC2Exception, [57](#)
  - FlyCapture2Managed::FC2Exception,  
[58](#)
- FC2Version, [59](#)
- FireBusReset
  - FlyCapture2Managed::Managed-  
BusManager, [77](#)
- FireSoftwareTrigger
  - FlyCapture2Managed::Managed-  
CameraBase, [96](#)
- FlyCapture2, [29](#)
- FlyCapture2Managed, [29](#)
  - AsyncCommandCallback, [34](#)
  - CommandCallbackDelegate, [34](#)
  - EnumCallback, [34](#)
  - ImageCallbackDelegate, [34](#)
  - ImageEventCallback, [34](#)
  - ManagedCameraEventCallback, [34](#)
  - ManagedCameraEventCallback-  
Delegate, [35](#)
  - htonl, [34](#)

- FlyCapture2Managed::BMPOption
  - BMPOption, [37](#)
  - indexedColor\_8bit, [37](#)
- FlyCapture2Managed::CameraInfo
  - applicationIPAddress, [40](#)
  - applicationPort, [40](#)
  - bayerTileFormat, [40](#)
  - busNumber, [40](#)
  - ccpStatus, [40](#)
  - configROM, [40](#)
  - defaultGateway, [41](#)
  - driverName, [41](#)
  - driverType, [41](#)
  - firmwareBuildTime, [41](#)
  - firmwareVersion, [41](#)
  - gigEMajorVersion, [41](#)
  - gigEMinorVersion, [41](#)
  - iidcVersion, [41](#)
  - interfaceType, [41](#)
  - ipAddress, [41](#)
  - isColorCamera, [42](#)
  - macAddress, [42](#)
  - maximumBusSpeed, [42](#)
  - modelName, [42](#)
  - nodeNumber, [42](#)
  - pcieBusSpeed, [42](#)
  - sensorInfo, [42](#)
  - sensorResolution, [42](#)
  - serialNumber, [42](#)
  - subnetMask, [42](#)
  - userDefinedName, [43](#)
  - vendorName, [43](#)
  - xmlURL1, [43](#)
  - xmlURL2, [43](#)
- FlyCapture2Managed::CameraProperty
  - CameraProperty, [44](#)
  - absControl, [44](#)
  - absValue, [44](#)
  - autoManualMode, [44](#)
  - onOff, [44](#)
  - onePush, [44](#)
  - present, [44](#)
  - type, [45](#)
  - valueA, [45](#)
  - valueB, [45](#)
- FlyCapture2Managed::CameraProperty-Info
  - CameraPropertyInfo, [46](#)
  - absMax, [46](#)
  - absMin, [46](#)
  - absValSupported, [46](#)
  - autoSupported, [46](#)
  - manualSupported, [47](#)
  - max, [47](#)
  - min, [47](#)
  - onOffSupported, [47](#)
  - onePushSupported, [47](#)
  - present, [47](#)
  - readOutSupported, [47](#)
  - type, [47](#)
  - unitAbbr, [47](#)
  - units, [47](#)
- FlyCapture2Managed::CameraStats
  - CameraStats, [50](#)
  - cameraCurrents, [50](#)
  - cameraPowerUp, [50](#)
  - cameraVoltages, [50](#)
  - imageCorrupt, [50](#)
  - imageDriverDropped, [50](#)
  - imageDropped, [50](#)
  - imageXmitFailed, [50](#)
  - numCurrents, [50](#)
  - numResendPacketsReceived, [50](#)
  - numResendPacketsRequested, [50](#)
  - numVoltages, [50](#)
  - portErrors, [50](#)
  - regReadFailed, [50](#)
  - regWriteFailed, [50](#)
  - temperature, [50](#)
  - timeSinceBusReset, [51](#)
  - timeSinceInitialization, [51](#)
  - timeStamp, [51](#)
- FlyCapture2Managed::ConfigROM
  - chipIdHi, [52](#)
  - chipIdLo, [52](#)
  - keyword, [52](#)
  - nodeVendorId, [52](#)
  - unitSWVer, [52](#)
  - unitSpecId, [52](#)
  - unitSubSWVer, [52](#)
  - vendorUniqueInfo0, [52](#)
  - vendorUniqueInfo1, [52](#)
  - vendorUniqueInfo2, [52](#)
  - vendorUniqueInfo3, [52](#)
- FlyCapture2Managed::EmbeddedImage-Info
  - EmbeddedImageInfo, [53](#)
  - GPIOPinState, [53](#)
  - ROIPosition, [54](#)
  - brightness, [53](#)

- exposure, 53
- frameCounter, 53
- gain, 53
- shutter, 54
- strobePattern, 54
- timestamp, 54
- whiteBalance, 54
- FlyCapture2Managed::EmbeddedImage-InfoProperty
  - available, 54
  - onOff, 54
- FlyCapture2Managed::FC2Config
  - FC2Config, 55
  - asyncBusSpeed, 55
  - bandwidthAllocation, 56
  - grabMode, 56
  - grabTimeout, 56
  - highPerformanceRetrieveBuffer, 56
  - isochBusSpeed, 56
  - minNumImageNotifications, 56
  - numBuffers, 56
  - numImageNotifications, 57
  - registerTimeout, 57
  - registerTimeoutRetries, 57
- FlyCapture2Managed::FC2Exception
  - ~FC2Exception, 58
  - CauseType, 58
  - FC2Exception, 58
  - NativeErrorTrace, 58
  - Type, 58
- FlyCapture2Managed::FC2Version
  - build, 59
  - major, 59
  - minor, 59
  - type, 59
- FlyCapture2Managed::Format7Image-Settings
  - height, 60
  - mode, 60
  - offsetX, 60
  - offsetY, 60
  - pixelFormat, 60
  - width, 60
- FlyCapture2Managed::Format7Info
  - imageHStepSize, 62
  - imageVStepSize, 62
  - maxHeight, 62
  - maxPacketSize, 62
  - maxWidth, 62
  - minPacketSize, 62
  - mode, 62
  - offsetHStepSize, 62
  - offsetVStepSize, 62
  - packetSize, 62
  - percentage, 62
  - pixelFormatBitField, 63
  - vendorPixelFormatBitField, 63
- FlyCapture2Managed::Format7PacketInfo
  - maxBytesPerPacket, 63
  - recommendedBytesPerPacket, 63
  - unitBytesPerPacket, 63
- FlyCapture2Managed::GigEConfig
  - enablePacketResend, 64
- FlyCapture2Managed::GigEImage-Settings
  - height, 65
  - offsetX, 65
  - offsetY, 65
  - pixelFormat, 65
  - width, 65
- FlyCapture2Managed::GigEImage-SettingsInfo
  - imageHStepSize, 66
  - imageVStepSize, 66
  - maxHeight, 66
  - maxWidth, 66
  - offsetHStepSize, 66
  - offsetVStepSize, 66
  - pixelFormatBitField, 66
  - vendorPixelFormatBitField, 66
- FlyCapture2Managed::GigEProperty
  - isReadable, 67
  - isWritable, 67
  - max, 67
  - min, 67
  - propType, 67
  - value, 68
- FlyCapture2Managed::GigEStream-Channel
  - destinationIpAddress, 68
  - doNotFragment, 68
  - hostPort, 69
  - interPacketDelay, 69
  - networkInterfaceIndex, 69
  - packetSize, 69
  - sourcePort, 69
- FlyCapture2Managed::Gui, 35
- FlyCapture2Managed::Gui::Camera-ControlDialog
  - ~CameraControlDialog, 38

- CameraControlDialog, 38
- Connect, 38
- Disconnect, 38
- Hide, 38
- IsVisible, 38
- SetTitle, 38
- Show, 38
- FlyCapture2Managed::Gui::Camera-  
SelectionDialog
  - ~CameraSelectionDialog, 48
  - CameraSelectionDialog, 48
  - GetSelectedCameraGuids, 48
  - SetTitle, 48
  - ShowModal, 49
- FlyCapture2Managed::ImageMetadata
  - embeddedBrightness, 70
  - embeddedExposure, 70
  - embeddedFrameCounter, 70
  - embeddedGPIOPinState, 70
  - embeddedGain, 70
  - embeddedROIPosition, 70
  - embeddedShutter, 70
  - embeddedStrobePattern, 71
  - embeddedTimeStamp, 71
  - embeddedWhiteBalance, 71
- FlyCapture2Managed::JpegOption
  - JpegOption, 71
  - progressive, 71
  - quality, 72
- FlyCapture2Managed::Jpg2Option
  - Jpg2Option, 72
  - quality, 72
- FlyCapture2Managed::LutData
  - enabled, 73
  - inputBitDepth, 73
  - numBanks, 73
  - numChannels, 73
  - numEntries, 73
  - outputBitDepth, 74
  - supported, 74
- FlyCapture2Managed::ManagedBus-  
Manager
  - ~ManagedBusManager, 76
  - ConvertToManagedGuid, 76
  - ConvertToNativeGuid, 76
  - DiscoverGigECameras, 77
  - FireBusReset, 77
  - ForceAllIPAddressesAutomatically,  
77
  - ForceIPAddressToCamera, 77
  - GetCameraFromIPAddress, 78
  - GetCameraFromIndex, 78
  - GetCameraFromSerialNumber, 78
  - GetCameraSerialNumberFromIndex,  
79
  - GetDeviceFromIndex, 79
  - GetInterfaceTypeFromGuid, 80
  - GetNumOfCameras, 80
  - GetNumOfDevices, 80
  - GetTopology, 80
  - GetUsbLinkInfo, 80
  - GetUsbPortStatus, 81
  - IsCameraControlable, 81
  - ManagedBusManager, 76
  - ReadPhyRegister, 81
  - RegisterCallback, 82
  - RescanBus, 82
  - UnregisterCallback, 82
  - WritePhyRegister, 83
- FlyCapture2Managed::ManagedCamera
  - ~ManagedCamera, 85
  - Connect, 86
  - GetFormat7Configuration, 86
  - GetFormat7Info, 86
  - GetVideoModeAndFrameRate, 87
  - GetVideoModeAndFrameRateInfo,  
87
  - ManagedCamera, 85
  - SetFormat7Configuration, 88
  - SetVideoModeAndFrameRate, 88
  - StartSyncCapture, 89
  - ValidateFormat7Settings, 90
- FlyCapture2Managed::ManagedCamera-  
Base
  - ~ManagedCameraBase, 95
  - Connect, 95
  - DeregisterAllEvents, 95
  - DeregisterEvent, 95
  - Disconnect, 95
  - EnableLUT, 96
  - FireSoftwareTrigger, 96
  - GetActiveLUTBank, 96
  - GetCameraInfo, 96
  - GetConfiguration, 97
  - GetCycleTime, 97
  - GetEmbeddedImageInfo, 97
  - GetGPIOPinDirection, 97
  - GetLUTBankInfo, 98
  - GetLUTChannel, 98
  - GetLUTInfo, 98

GetMemoryChannel, [99](#)  
GetMemoryChannelInfo, [99](#)  
GetNativeCamera, [99](#)  
GetProperty, [99](#)  
GetPropertyInfo, [100](#)  
GetRegisterString, [100](#)  
GetStats, [100](#)  
GetStrobe, [100](#)  
GetStrobeInfo, [101](#)  
GetTriggerDelay, [101](#)  
GetTriggerDelayInfo, [102](#)  
GetTriggerMode, [102](#)  
GetTriggerModeInfo, [102](#)  
IsConnected, [102](#)  
ManagedCameraBase, [95](#)  
OnNativeCallback, [103](#)  
OnNativeCameraEventCallback, [103](#)  
ReadRegister, [103](#)  
ReadRegisterBlock, [103](#)  
RegisterAllEvents, [104](#)  
RegisterEvent, [104](#)  
ResetStats, [104](#)  
RestoreFromMemoryChannel, [104](#)  
RetrieveBuffer, [104](#)  
SaveToMemoryChannel, [104](#)  
SetActiveLUTBank, [105](#)  
SetCallback, [105](#)  
SetCamera, [105](#)  
SetConfiguration, [105](#)  
SetEmbeddedImageInfo, [106](#)  
SetGPIOPinDirection, [106](#)  
SetLUTChannel, [107](#)  
SetProperty, [107](#), [108](#)  
SetStrobe, [108](#)  
SetTriggerDelay, [108](#), [109](#)  
SetTriggerMode, [109](#)  
SetUserBuffers, [109](#)  
StartCapture, [110](#)  
StopCapture, [111](#)  
WaitForBufferEvent, [111](#)  
WriteRegister, [111](#), [112](#)  
WriteRegisterBlock, [112](#)  
m\_allInternalCameraEvents, [112](#)  
m\_externalDelegate, [112](#)  
m\_internalCameraEventDelegate, [112](#)  
m\_internalDelegate, [112](#)  
m\_isLocal, [112](#)  
m\_p, [113](#)  
m\_pNativeCamBase, [113](#)  
m\_specificInternalCameraEvents, [113](#)  
FlyCapture2Managed::ManagedEvent-  
    CallbackData  
    EventID, [113](#)  
    EventName, [113](#)  
    EventTimestamp, [113](#)  
FlyCapture2Managed::ManagedEvent-  
    Options  
    EventCallbackFcn, [114](#)  
    EventName, [114](#)  
FlyCapture2Managed::ManagedGC-  
    Camera  
    ~ManagedGCCamera, [115](#)  
    Connect, [115](#), [116](#)  
    Disconnect, [116](#)  
    GetNodeMap, [116](#)  
    ManagedGCCamera, [115](#)  
    SetCamera, [116](#)  
FlyCapture2Managed::ManagedGCPort  
    ~ManagedGCPort, [117](#)  
    ManagedGCPort, [117](#)  
    Read, [117](#)  
    Write, [117](#)  
FlyCapture2Managed::ManagedGigE-  
    Camera  
    ~ManagedGigECamera, [120](#)  
    Connect, [120](#)  
    DiscoverGigEPacketSize, [121](#)  
    GetGigEConfig, [121](#)  
    GetGigEImageBinningSettings, [121](#)  
    GetGigEImageSettings, [122](#)  
    GetGigEImageSettingsInfo, [122](#)  
    GetGigEImagingMode, [122](#)  
    GetGigEProperty, [122](#)  
    GetGigEStreamChannelInfo, [122](#)  
    GetNumStreamChannels, [122](#)  
    ManagedGigECamera, [120](#)  
    QueryGigEImagingMode, [123](#)  
    ReadGVCPMemory, [123](#)  
    ReadGVCPRegister, [123](#)  
    ReadGVCPRegisterBlock, [123](#)  
    SetGigEConfig, [124](#)  
    SetGigEImageBinningSettings, [124](#)  
    SetGigEImageSettings, [124](#)  
    SetGigEImagingMode, [124](#)  
    SetGigEProperty, [125](#)  
    SetGigEStreamChannelInfo, [125](#)  
    WriteGVCPMemory, [125](#)  
    WriteGVCPRegister, [125](#)

- WriteGVCPRegisterBlock, 126
- FlyCapture2Managed::ManagedImage
  - ~ManagedImage, 129
  - CalculateStatistics, 129
  - Convert, 130
  - DetermineBitsPerPixel, 130
  - GetDimensions, 131
  - GetNativeImage, 131
  - GetRawNativeImagePointer, 131
  - IsNativeImageValid, 131
  - ManagedImage, 129
  - ReleaseBuffer, 131
  - Save, 131–133
  - SetData, 133
  - SetDimensions, 133
  - bayerTileFormat, 133
  - bitmap, 134
  - bitsPerPixel, 134
  - blockId, 134
  - colorProcessingAlgorithm, 134
  - cols, 134
  - data, 134
  - dataSize, 134
  - defaultColorProcessingAlgorithm, 134
  - defaultOutputPixelFormat, 134
  - imageMetadata, 135
  - pixelFormat, 135
  - receivedDataSize, 135
  - rows, 135
  - stride, 135
  - timeStamp, 135
- FlyCapture2Managed::ManagedImage-Statistics
  - ~ManagedImageStatistics, 136
  - DisableAll, 136
  - EnableAll, 136
  - EnableGreyOnly, 136
  - EnableHSLOnly, 136
  - EnableRGBOnly, 136
  - GetChannelStatus, 137
  - GetHistogram, 137
  - GetMean, 137
  - GetNativeImageStatistics, 137
  - GetNumPixelValues, 137
  - GetPixelValueRange, 137
  - GetRange, 137
  - GetStatistics, 137
  - ManagedImageStatistics, 136
  - SetChannelStatus, 137
- FlyCapture2Managed::ManagedPGR-Guid
  - Equals, 138
  - GetHashCode, 138
  - ManagedPGRGuid, 138
  - operator=, 138
  - operator==, 139
  - value0, 139
  - value1, 139
  - value2, 139
  - value3, 139
- FlyCapture2Managed::ManagedTopology-Node
  - ~ManagedTopologyNode, 141
  - GetChild, 141
  - GetDeviceId, 141
  - GetGuid, 141
  - GetInterfaceType, 142
  - GetNodeType, 142
  - GetNumChildren, 142
  - GetNumPorts, 142
  - GetPortType, 142
  - ManagedTopologyNode, 141
  - NodeType, 140
  - PortType, 140
  - TranslateNodeType, 143
  - TranslatePortType, 143
- FlyCapture2Managed::ManagedUtilities
  - CheckDriver, 144
  - GetDriverDeviceName, 144
  - LaunchBrowser, 144
  - LaunchCommand, 144
  - LaunchCommandAsync, 144
  - LaunchHelp, 144
  - OnNativeCallback, 144
  - libraryVersion, 144
  - m\_externalDelegate, 144
  - m\_internalDelegate, 144
  - systemInfo, 144
- FlyCapture2Managed::NativeEventStruct ptr, 144
- FlyCapture2Managed::PgmOption
  - PgmOption, 145
  - binaryFile, 145
- FlyCapture2Managed::PngOption
  - PngOption, 146
  - compressionLevel, 146
  - interlaced, 146
- FlyCapture2Managed::PpmOption
  - PpmOption, 146



- binaryFile, 146
- FlyCapture2Managed::StrobeControl
  - delay, 147
  - duration, 147
  - onOff, 147
  - polarity, 147
  - source, 147
- FlyCapture2Managed::StrobeInfo
  - maxValue, 148
  - minValue, 148
  - onOffSupported, 148
  - polaritySupported, 148
  - present, 148
  - readOutSupported, 149
  - source, 149
- FlyCapture2Managed::SystemInfo
  - byteOrder, 150
  - cpuDescription, 150
  - driverList, 150
  - gpuDescription, 150
  - libraryList, 150
  - numCpuCores, 150
  - osDescription, 150
  - osType, 150
  - screenHeight, 150
  - screenWidth, 150
  - systemMemorySize, 151
- FlyCapture2Managed::TiffOption
  - CompressionMethod, 151
  - TiffOption, 152
  - compression, 152
- FlyCapture2Managed::TimeStamp
  - cycleCount, 152
  - cycleOffset, 152
  - cycleSeconds, 153
  - microSeconds, 153
  - seconds, 153
- FlyCapture2Managed::Translate
  - ToMgd, 156–158
  - ToNative, 158–160
  - Translate::ToMgd, 162
  - Translate::ToNative, 162
  - translate, 160–162
- FlyCapture2Managed::TriggerMode
  - mode, 163
  - onOff, 163
  - parameter, 163
  - polarity, 163
  - source, 163
- FlyCapture2Managed::TriggerModelInfo
  - modeMask, 164
  - onOffSupported, 164
  - polaritySupported, 164
  - present, 164
  - readOutSupported, 165
  - softwareTriggerSupported, 165
  - sourceMask, 165
  - valueReadable, 165
- ForceAllIPAddressesAutomatically
  - FlyCapture2Managed::Managed-BusManager, 77
- ForceIPAddressToCamera
  - FlyCapture2Managed::Managed-BusManager, 77
- Format7ImageSettings, 59
- Format7Info, 61
- Format7PacketInfo, 63
- FrameRate
  - Enumerations, 19
- GPIOPinState
  - FlyCapture2Managed::Embedded-ImageInfo, 53
- GetActiveLUTBank
  - FlyCapture2Managed::Managed-CameraBase, 96
- GetCameraFromIPAddress
  - FlyCapture2Managed::Managed-BusManager, 78
- GetCameraFromIndex
  - FlyCapture2Managed::Managed-BusManager, 78
- GetCameraFromSerialNumber
  - FlyCapture2Managed::Managed-BusManager, 78
- GetCameraInfo
  - FlyCapture2Managed::Managed-CameraBase, 96
- GetCameraSerialNumberFromIndex
  - FlyCapture2Managed::Managed-BusManager, 79
- GetChannelStatus
  - FlyCapture2Managed::Managed-ImageStatistics, 137
- GetChild
  - FlyCapture2Managed::Managed-TopologyNode, 141
- GetConfiguration
  - FlyCapture2Managed::Managed-CameraBase, 97
- GetCycleTime

- FlyCapture2Managed::Managed-CameraBase, [97](#)
- GetDeviceFromIndex
  - FlyCapture2Managed::Managed-BusManager, [79](#)
- GetDeviceId
  - FlyCapture2Managed::Managed-TopologyNode, [141](#)
- GetDimensions
  - FlyCapture2Managed::Managed-Image, [131](#)
- GetDriverDeviceName
  - FlyCapture2Managed::Managed-Utilities, [144](#)
- GetEmbeddedImageInfo
  - FlyCapture2Managed::Managed-CameraBase, [97](#)
- GetFormat7Configuration
  - FlyCapture2Managed::Managed-Camera, [86](#)
- GetFormat7Info
  - FlyCapture2Managed::Managed-Camera, [86](#)
- GetGPIOPinDirection
  - FlyCapture2Managed::Managed-CameraBase, [97](#)
- GetGigEConfig
  - FlyCapture2Managed::ManagedGigECamera, [121](#)
- GetGigEImageBinningSettings
  - FlyCapture2Managed::ManagedGigECamera, [121](#)
- GetGigEImageSettings
  - FlyCapture2Managed::ManagedGigECamera, [122](#)
- GetGigEImageSettingsInfo
  - FlyCapture2Managed::ManagedGigECamera, [122](#)
- GetGigEImagingMode
  - FlyCapture2Managed::ManagedGigECamera, [122](#)
- GetGigEProperty
  - FlyCapture2Managed::ManagedGigECamera, [122](#)
- GetGigEStreamChannelInfo
  - FlyCapture2Managed::ManagedGigECamera, [122](#)
- GetGuid
  - FlyCapture2Managed::Managed-TopologyNode, [141](#)
- GetHashCode
  - FlyCapture2Managed::ManagedPG-RGuid, [138](#)
- GetHistogram
  - FlyCapture2Managed::Managed-ImageStatistics, [137](#)
- GetInterfaceType
  - FlyCapture2Managed::Managed-TopologyNode, [142](#)
- GetInterfaceTypeFromGuid
  - FlyCapture2Managed::Managed-BusManager, [80](#)
- GetLUTBankInfo
  - FlyCapture2Managed::Managed-CameraBase, [98](#)
- GetLUTChannel
  - FlyCapture2Managed::Managed-CameraBase, [98](#)
- GetLUTInfo
  - FlyCapture2Managed::Managed-CameraBase, [98](#)
- GetMean
  - FlyCapture2Managed::Managed-ImageStatistics, [137](#)
- GetMemoryChannel
  - FlyCapture2Managed::Managed-CameraBase, [99](#)
- GetMemoryChannelInfo
  - FlyCapture2Managed::Managed-CameraBase, [99](#)
- GetNativeCamera
  - FlyCapture2Managed::Managed-CameraBase, [99](#)
- GetNativeImage
  - FlyCapture2Managed::Managed-Image, [131](#)
- GetNativeImageStatistics
  - FlyCapture2Managed::Managed-ImageStatistics, [137](#)
- GetNodeMap
  - FlyCapture2Managed::ManagedGC-Camera, [116](#)
- GetNodeType
  - FlyCapture2Managed::Managed-TopologyNode, [142](#)
- GetNumChildren
  - FlyCapture2Managed::Managed-TopologyNode, [142](#)
- GetNumOfCameras

- FlyCapture2Managed::Managed-BusManager, [80](#)
- GetNumOfDevices
  - FlyCapture2Managed::Managed-BusManager, [80](#)
- GetNumPixelValues
  - FlyCapture2Managed::Managed-ImageStatistics, [137](#)
- GetNumPorts
  - FlyCapture2Managed::Managed-TopologyNode, [142](#)
- GetNumStreamChannels
  - FlyCapture2Managed::ManagedGigECamera, [122](#)
- GetPixelValueRange
  - FlyCapture2Managed::Managed-ImageStatistics, [137](#)
- GetPortType
  - FlyCapture2Managed::Managed-TopologyNode, [142](#)
- GetProperty
  - FlyCapture2Managed::Managed-CameraBase, [99](#)
- GetPropertyInfo
  - FlyCapture2Managed::Managed-CameraBase, [100](#)
- GetRange
  - FlyCapture2Managed::Managed-ImageStatistics, [137](#)
- GetRawNativeImagePointer
  - FlyCapture2Managed::Managed-Image, [131](#)
- GetRegisterString
  - FlyCapture2Managed::Managed-CameraBase, [100](#)
- GetSelectedCameraGuids
  - FlyCapture2Managed::Gui::Camera-SelectionDialog, [48](#)
- GetStatistics
  - FlyCapture2Managed::Managed-ImageStatistics, [137](#)
- GetStats
  - FlyCapture2Managed::Managed-CameraBase, [100](#)
- GetStrobe
  - FlyCapture2Managed::Managed-CameraBase, [100](#)
- GetStrobeInfo
  - FlyCapture2Managed::Managed-CameraBase, [101](#)
- GetTopology
  - FlyCapture2Managed::Managed-BusManager, [80](#)
- GetTriggerDelay
  - FlyCapture2Managed::Managed-CameraBase, [101](#)
- GetTriggerDelayInfo
  - FlyCapture2Managed::Managed-CameraBase, [102](#)
- GetTriggerMode
  - FlyCapture2Managed::Managed-CameraBase, [102](#)
- GetTriggerModelInfo
  - FlyCapture2Managed::Managed-CameraBase, [102](#)
- GetUsbLinkInfo
  - FlyCapture2Managed::Managed-BusManager, [80](#)
- GetUsbPortStatus
  - FlyCapture2Managed::Managed-BusManager, [81](#)
- GetVideoModeAndFrameRate
  - FlyCapture2Managed::Managed-Camera, [87](#)
- GetVideoModeAndFrameRateInfo
  - FlyCapture2Managed::Managed-Camera, [87](#)
- GigEConfig, [64](#)
- GigEImageSettings, [64](#)
- GigEImageSettingsInfo, [65](#)
- GigEProperty, [67](#)
- GigEPropertyType
  - Enumerations, [19](#)
- GigEStreamChannel, [68](#)
- GrabMode
  - Enumerations, [19](#)
- GrabTimeout
  - Enumerations, [20](#)
- Hide
  - FlyCapture2Managed::Gui::Camera-ControlDialog, [38](#)
- Image saving structures., [28](#)
- ImageCallbackDelegate
  - FlyCapture2Managed, [34](#)
- ImageEventCallback
  - FlyCapture2Managed, [34](#)
- ImageFileFormat
  - Enumerations, [20](#)
- ImageMetadata, [69](#)
- InterfaceType

- Enumerations, [21](#)
- IsCameraControlable
  - FlyCapture2Managed::Managed-BusManager, [81](#)
- IsConnected
  - FlyCapture2Managed::Managed-CameraBase, [102](#)
- IsNativeImageValid
  - FlyCapture2Managed::Managed-Image, [131](#)
- IsVisible
  - FlyCapture2Managed::Gui::Camera-ControlDialog, [38](#)
- JpegOption, [71](#)
  - FlyCapture2Managed::JpegOption, [71](#)
- Jpg2Option, [72](#)
  - FlyCapture2Managed::Jpg2Option, [72](#)
- LaunchBrowser
  - FlyCapture2Managed::Managed-Utilities, [144](#)
- LaunchCommand
  - FlyCapture2Managed::Managed-Utilities, [144](#)
- LaunchCommandAsync
  - FlyCapture2Managed::Managed-Utilities, [144](#)
- LaunchHelp
  - FlyCapture2Managed::Managed-Utilities, [144](#)
- LutData, [73](#)
- ManagedBusManager, [74](#)
  - FlyCapture2Managed::Managed-BusManager, [76](#)
- ManagedCallbackType
  - Enumerations, [21](#)
- ManagedCamera, [83](#)
  - FlyCapture2Managed::Managed-Camera, [85](#)
- ManagedCameraBase, [90](#)
  - FlyCapture2Managed::Managed-CameraBase, [95](#)
- ManagedCameraEventCallback
  - FlyCapture2Managed, [34](#)
- ManagedCameraEventCallbackDelegate
  - FlyCapture2Managed, [35](#)
- ManagedEventCallbackData, [113](#)
- ManagedEventOptions, [114](#)
- ManagedGCCamera, [114](#)
  - FlyCapture2Managed::ManagedGC-Camera, [115](#)
- ManagedGCPort, [116](#)
  - FlyCapture2Managed::ManagedGC-Port, [117](#)
- ManagedGigECamera, [117](#)
  - FlyCapture2Managed::ManagedGig-ECamera, [120](#)
- ManagedImage, [126](#)
  - FlyCapture2Managed::Managed-Image, [129](#)
- ManagedImageStatistics, [136](#)
  - FlyCapture2Managed::Managed-ImageStatistics, [136](#)
- ManagedPGRGuid, [137](#)
  - FlyCapture2Managed::ManagedPG-RGuid, [138](#)
- ManagedTopologyNode, [139](#)
  - FlyCapture2Managed::Managed-TopologyNode, [141](#)
- ManagedUtilities, [143](#)
- Mode
  - Enumerations, [21](#)
- NativeErrorTrace
  - FlyCapture2Managed::FC2Exception, [58](#)
- NativeEventStruct, [144](#)
- NodeType
  - FlyCapture2Managed::Managed-TopologyNode, [140](#)
- OSType
  - Enumerations, [22](#)
- OnNativeCallback
  - FlyCapture2Managed::Managed-CameraBase, [103](#)
  - FlyCapture2Managed::Managed-Utilities, [144](#)
- OnNativeCameraEventCallback
  - FlyCapture2Managed::Managed-CameraBase, [103](#)
- PCleBusSpeed
  - Enumerations, [22](#)
- PgmOption, [144](#)
  - FlyCapture2Managed::PgmOption, [145](#)
- PixelFormat
  - Enumerations, [23](#)
- PngOption, [145](#)
  - FlyCapture2Managed::PngOption, [146](#)

- PortType
  - FlyCapture2Managed::Managed-TopologyNode, [140](#)
- PpmOption, [146](#)
  - FlyCapture2Managed::PpmOption, [146](#)
- PropertyType
  - Enumerations, [23](#)
- QueryGigEImagingMode
  - FlyCapture2Managed::ManagedGig-ECamera, [123](#)
- ROIPosition
  - FlyCapture2Managed::Embedded-ImageInfo, [54](#)
- Read
  - FlyCapture2Managed::ManagedGC-Port, [117](#)
- ReadGVCPMemory
  - FlyCapture2Managed::ManagedGig-ECamera, [123](#)
- ReadGVCPRegister
  - FlyCapture2Managed::ManagedGig-ECamera, [123](#)
- ReadGVCPRegisterBlock
  - FlyCapture2Managed::ManagedGig-ECamera, [123](#)
- ReadPhyRegister
  - FlyCapture2Managed::Managed-BusManager, [81](#)
- ReadRegister
  - FlyCapture2Managed::Managed-CameraBase, [103](#)
- ReadRegisterBlock
  - FlyCapture2Managed::Managed-CameraBase, [103](#)
- RegisterAllEvents
  - FlyCapture2Managed::Managed-CameraBase, [104](#)
- RegisterCallback
  - FlyCapture2Managed::Managed-BusManager, [82](#)
- RegisterEvent
  - FlyCapture2Managed::Managed-CameraBase, [104](#)
- ReleaseBuffer
  - FlyCapture2Managed::Managed-Image, [131](#)
- RescanBus
  - FlyCapture2Managed::Managed-BusManager, [82](#)
- ResetStats
  - FlyCapture2Managed::Managed-CameraBase, [104](#)
- RestoreFromMemoryChannel
  - FlyCapture2Managed::Managed-CameraBase, [104](#)
- RetrieveBuffer
  - FlyCapture2Managed::Managed-CameraBase, [104](#)
- Save
  - FlyCapture2Managed::Managed-Image, [131–133](#)
- SaveToMemoryChannel
  - FlyCapture2Managed::Managed-CameraBase, [104](#)
- SetActiveLUTBank
  - FlyCapture2Managed::Managed-CameraBase, [105](#)
- SetCallback
  - FlyCapture2Managed::Managed-CameraBase, [105](#)
- SetCamera
  - FlyCapture2Managed::Managed-CameraBase, [105](#)
  - FlyCapture2Managed::ManagedGC-Camera, [116](#)
- SetChannelStatus
  - FlyCapture2Managed::Managed-ImageStatistics, [137](#)
- SetConfiguration
  - FlyCapture2Managed::Managed-CameraBase, [105](#)
- SetData
  - FlyCapture2Managed::Managed-Image, [133](#)
- SetDimensions
  - FlyCapture2Managed::Managed-Image, [133](#)
- SetEmbeddedImageInfo
  - FlyCapture2Managed::Managed-CameraBase, [106](#)
- SetFormat7Configuration
  - FlyCapture2Managed::Managed-Camera, [88](#)
- SetGPIOPinDirection
  - FlyCapture2Managed::Managed-CameraBase, [106](#)
- SetGigEConfig
  - FlyCapture2Managed::ManagedGig-ECamera, [124](#)

- SetGigEImageBinningSettings
  - FlyCapture2Managed::ManagedGig-ECamera, [124](#)
- SetGigEImageSettings
  - FlyCapture2Managed::ManagedGig-ECamera, [124](#)
- SetGigEImagingMode
  - FlyCapture2Managed::ManagedGig-ECamera, [124](#)
- SetGigEProperty
  - FlyCapture2Managed::ManagedGig-ECamera, [125](#)
- SetGigEStreamChannelInfo
  - FlyCapture2Managed::ManagedGig-ECamera, [125](#)
- SetLUTChannel
  - FlyCapture2Managed::Managed-CameraBase, [107](#)
- SetProperty
  - FlyCapture2Managed::Managed-CameraBase, [107](#), [108](#)
- SetStrobe
  - FlyCapture2Managed::Managed-CameraBase, [108](#)
- SetTitle
  - FlyCapture2Managed::Gui::Camera-ControlDialog, [38](#)
  - FlyCapture2Managed::Gui::Camera-SelectionDialog, [48](#)
- SetTriggerDelay
  - FlyCapture2Managed::Managed-CameraBase, [108](#), [109](#)
- SetTriggerMode
  - FlyCapture2Managed::Managed-CameraBase, [109](#)
- SetUserBuffers
  - FlyCapture2Managed::Managed-CameraBase, [109](#)
- SetVideoModeAndFrameRate
  - FlyCapture2Managed::Managed-Camera, [88](#)
- Show
  - FlyCapture2Managed::Gui::Camera-ControlDialog, [38](#)
- ShowModal
  - FlyCapture2Managed::Gui::Camera-SelectionDialog, [49](#)
- StartCapture
  - FlyCapture2Managed::Managed-CameraBase, [110](#)
- StartSyncCapture
  - FlyCapture2Managed::Managed-Camera, [89](#)
- StatisticsChannel
  - Enumerations, [24](#)
- StopCapture
  - FlyCapture2Managed::Managed-CameraBase, [111](#)
- StrobeControl, [147](#)
- StrobeInfo, [148](#)
- Structures, [26](#)
- SystemInfo, [149](#)
- TiffOption, [151](#)
  - FlyCapture2Managed::TiffOption, [152](#)
- TimeStamp, [152](#)
- ToMgd
  - FlyCapture2Managed::Translate, [156–158](#)
- ToNative
  - FlyCapture2Managed::Translate, [158–160](#)
- Translate, [153](#)
- Translate::ToMgd
  - FlyCapture2Managed::Translate, [162](#)
- Translate::ToNative
  - FlyCapture2Managed::Translate, [162](#)
- TranslateNodeType
  - FlyCapture2Managed::Managed-TopologyNode, [143](#)
- TranslatePortType
  - FlyCapture2Managed::Managed-TopologyNode, [143](#)
- TriggerMode, [163](#)
- TriggerModeInfo, [164](#)
- Type
  - FlyCapture2Managed::FC2Exception, [58](#)
- UnregisterCallback
  - FlyCapture2Managed::Managed-BusManager, [82](#)
- ValidateFormat7Settings
  - FlyCapture2Managed::Managed-Camera, [90](#)
- VideoMode
  - Enumerations, [24](#)
- WaitForBufferEvent
  - FlyCapture2Managed::Managed-CameraBase, [111](#)
- Write

- FlyCapture2Managed::ManagedGC-Port, [117](#)
- WriteGVCPMemory
  - FlyCapture2Managed::ManagedGig-ECamera, [125](#)
- WriteGVCPRegister
  - FlyCapture2Managed::ManagedGig-ECamera, [125](#)
- WriteGVCPRegisterBlock
  - FlyCapture2Managed::ManagedGig-ECamera, [126](#)
- WritePhyRegister
  - FlyCapture2Managed::Managed-BusManager, [83](#)
- WriteRegister
  - FlyCapture2Managed::Managed-CameraBase, [111](#), [112](#)
- WriteRegisterBlock
  - FlyCapture2Managed::Managed-CameraBase, [112](#)
- absControl
  - FlyCapture2Managed::Camera-Property, [44](#)
- absMax
  - FlyCapture2Managed::Camera-PropertyInfo, [46](#)
- absMin
  - FlyCapture2Managed::Camera-PropertyInfo, [46](#)
- absValSupported
  - FlyCapture2Managed::Camera-PropertyInfo, [46](#)
- absValue
  - FlyCapture2Managed::Camera-Property, [44](#)
- applicationIPAddress
  - FlyCapture2Managed::CameraInfo, [40](#)
- applicationPort
  - FlyCapture2Managed::CameraInfo, [40](#)
- asyncBusSpeed
  - FlyCapture2Managed::FC2Config, [55](#)
- autoManualMode
  - FlyCapture2Managed::Camera-Property, [44](#)
- autoSupported
  - FlyCapture2Managed::Camera-PropertyInfo, [46](#)
- available
  - FlyCapture2Managed::Embedded-ImageInfoProperty, [54](#)
- bandwidthAllocation
  - FlyCapture2Managed::FC2Config, [56](#)
- bayerTileFormat
  - FlyCapture2Managed::CameraInfo, [40](#)
  - FlyCapture2Managed::Managed-Image, [133](#)
- binaryFile
  - FlyCapture2Managed::PgmOption, [145](#)
  - FlyCapture2Managed::PpmOption, [146](#)
- bitmap
  - FlyCapture2Managed::Managed-Image, [134](#)
- bitsPerPixel
  - FlyCapture2Managed::Managed-Image, [134](#)
- blockId
  - FlyCapture2Managed::Managed-Image, [134](#)
- brightness
  - FlyCapture2Managed::Embedded-ImageInfo, [53](#)
- build
  - FlyCapture2Managed::FC2Version, [59](#)
- busNumber
  - FlyCapture2Managed::CameraInfo, [40](#)
- byteOrder
  - FlyCapture2Managed::SystemInfo, [150](#)
- cameraCurrents
  - FlyCapture2Managed::CameraStats, [50](#)
- cameraPowerUp
  - FlyCapture2Managed::CameraStats, [50](#)
- cameraVoltages
  - FlyCapture2Managed::CameraStats, [50](#)

- ccpStatus
  - FlyCapture2Managed::CameraInfo, [40](#)
- chipIdHi
  - FlyCapture2Managed::ConfigROM, [52](#)
- chipIdLo
  - FlyCapture2Managed::ConfigROM, [52](#)
- colorProcessingAlgorithm
  - FlyCapture2Managed::Managed-Image, [134](#)
- cols
  - FlyCapture2Managed::Managed-Image, [134](#)
- compression
  - FlyCapture2Managed::TiffOption, [152](#)
- compressionLevel
  - FlyCapture2Managed::PngOption, [146](#)
- configROM
  - FlyCapture2Managed::CameraInfo, [40](#)
- cpuDescription
  - FlyCapture2Managed::SystemInfo, [150](#)
- cycleCount
  - FlyCapture2Managed::TimeStamp, [152](#)
- cycleOffset
  - FlyCapture2Managed::TimeStamp, [152](#)
- cycleSeconds
  - FlyCapture2Managed::TimeStamp, [153](#)
- data
  - FlyCapture2Managed::Managed-Image, [134](#)
- dataSize
  - FlyCapture2Managed::Managed-Image, [134](#)
- defaultColorProcessingAlgorithm
  - FlyCapture2Managed::Managed-Image, [134](#)
- defaultGateway
  - FlyCapture2Managed::CameraInfo, [41](#)
- defaultOutputPixelFormat
  - FlyCapture2Managed::Managed-Image, [134](#)
- delay
  - FlyCapture2Managed::Strobe-Control, [147](#)
- destinationIpAddress
  - FlyCapture2Managed::GigEStream-Channel, [68](#)
- doNotFragment
  - FlyCapture2Managed::GigEStream-Channel, [68](#)
- driverList
  - FlyCapture2Managed::SystemInfo, [150](#)
- driverName
  - FlyCapture2Managed::CameraInfo, [41](#)
- driverType
  - FlyCapture2Managed::CameraInfo, [41](#)
- duration
  - FlyCapture2Managed::Strobe-Control, [147](#)
- embeddedBrightness
  - FlyCapture2Managed::Image-Metadata, [70](#)
- embeddedExposure
  - FlyCapture2Managed::Image-Metadata, [70](#)
- embeddedFrameCounter
  - FlyCapture2Managed::Image-Metadata, [70](#)
- embeddedGPIOPinState
  - FlyCapture2Managed::Image-Metadata, [70](#)
- embeddedGain
  - FlyCapture2Managed::Image-Metadata, [70](#)
- embeddedROIPosition
  - FlyCapture2Managed::Image-Metadata, [70](#)
- embeddedShutter
  - FlyCapture2Managed::Image-Metadata, [70](#)
- embeddedStrobePattern
  - FlyCapture2Managed::Image-Metadata, [71](#)
- embeddedTimeStamp
  - FlyCapture2Managed::Image-Metadata, [70](#)



- FlyCapture2Managed::Image-Metadata, [71](#)
- embeddedWhiteBalance
  - FlyCapture2Managed::Image-Metadata, [71](#)
- enablePacketResend
  - FlyCapture2Managed::GigEConfig, [64](#)
- enabled
  - FlyCapture2Managed::LutData, [73](#)
- exposure
  - FlyCapture2Managed::Embedded-ImageInfo, [53](#)
- firmwareBuildTime
  - FlyCapture2Managed::CameraInfo, [41](#)
- firmwareVersion
  - FlyCapture2Managed::CameraInfo, [41](#)
- frameCounter
  - FlyCapture2Managed::Embedded-ImageInfo, [53](#)
- gain
  - FlyCapture2Managed::Embedded-ImageInfo, [53](#)
- gigEMajorVersion
  - FlyCapture2Managed::CameraInfo, [41](#)
- gigEMinorVersion
  - FlyCapture2Managed::CameraInfo, [41](#)
- gpuDescription
  - FlyCapture2Managed::SystemInfo, [150](#)
- grabMode
  - FlyCapture2Managed::FC2Config, [56](#)
- grabTimeout
  - FlyCapture2Managed::FC2Config, [56](#)
- height
  - FlyCapture2Managed::Format7-ImageSettings, [60](#)
  - FlyCapture2Managed::GigEImageSettings, [65](#)
- highPerformanceRetrieveBuffer
  - FlyCapture2Managed::FC2Config, [56](#)
- hostPort
  - FlyCapture2Managed::GigEStream-Channel, [69](#)
- htonl
  - FlyCapture2Managed, [34](#)
- iidcVersion
  - FlyCapture2Managed::CameraInfo, [41](#)
- imageCorrupt
  - FlyCapture2Managed::CameraStats, [50](#)
- imageDriverDropped
  - FlyCapture2Managed::CameraStats, [50](#)
- imageDropped
  - FlyCapture2Managed::CameraStats, [50](#)
- imageHStepSize
  - FlyCapture2Managed::Format7Info, [62](#)
  - FlyCapture2Managed::GigEImageSettingsInfo, [66](#)
- imageMetadata
  - FlyCapture2Managed::Managed-Image, [135](#)
- imageVStepSize
  - FlyCapture2Managed::Format7Info, [62](#)
  - FlyCapture2Managed::GigEImageSettingsInfo, [66](#)
- imageXmitFailed
  - FlyCapture2Managed::CameraStats, [50](#)
- indexedColor\_8bit
  - FlyCapture2Managed::BMPOption, [37](#)
- inputBitDepth
  - FlyCapture2Managed::LutData, [73](#)
- interPacketDelay
  - FlyCapture2Managed::GigEStream-Channel, [69](#)
- interfaceType
  - FlyCapture2Managed::CameraInfo, [41](#)
- interlaced
  - FlyCapture2Managed::PngOption, [146](#)
- ipAddress

- FlyCapture2Managed::CameraInfo, 41
- isColorCamera
  - FlyCapture2Managed::CameraInfo, 42
- isReadable
  - FlyCapture2Managed::GigEProperty, 67
- isWritable
  - FlyCapture2Managed::GigEProperty, 67
- isochBusSpeed
  - FlyCapture2Managed::FC2Config, 56
- keyword
  - FlyCapture2Managed::ConfigROM, 52
- libraryList
  - FlyCapture2Managed::SystemInfo, 150
- libraryVersion
  - FlyCapture2Managed::Managed-Utilities, 144
- m\_allInternalCameraEvents
  - FlyCapture2Managed::Managed-CameraBase, 112
- m\_externalDelegate
  - FlyCapture2Managed::Managed-CameraBase, 112
  - FlyCapture2Managed::Managed-Utilities, 144
- m\_internalCameraEventDelegate
  - FlyCapture2Managed::Managed-CameraBase, 112
- m\_internalDelegate
  - FlyCapture2Managed::Managed-CameraBase, 112
  - FlyCapture2Managed::Managed-Utilities, 144
- m\_isLocal
  - FlyCapture2Managed::Managed-CameraBase, 112
- m\_p
  - FlyCapture2Managed::Managed-CameraBase, 113
- m\_pNativeCamBase
  - FlyCapture2Managed::Managed-CameraBase, 113
- m\_specificInternalCameraEvents
  - FlyCapture2Managed::Managed-CameraBase, 113
- macAddress
  - FlyCapture2Managed::CameraInfo, 42
- major
  - FlyCapture2Managed::FC2Version, 59
- manualSupported
  - FlyCapture2Managed::Camera-PropertyInfo, 47
- max
  - FlyCapture2Managed::Camera-PropertyInfo, 47
  - FlyCapture2Managed::GigEProperty, 67
- maxBytesPerPacket
  - FlyCapture2Managed::Format7-PacketInfo, 63
- maxHeight
  - FlyCapture2Managed::Format7Info, 62
  - FlyCapture2Managed::GigEImage-SettingsInfo, 66
- maxPacketSize
  - FlyCapture2Managed::Format7Info, 62
- maxValue
  - FlyCapture2Managed::StrobeInfo, 148
- maxWidth
  - FlyCapture2Managed::Format7Info, 62
  - FlyCapture2Managed::GigEImage-SettingsInfo, 66
- maximumBusSpeed
  - FlyCapture2Managed::CameraInfo, 42
- microSeconds
  - FlyCapture2Managed::TimeStamp, 153
- min
  - FlyCapture2Managed::Camera-PropertyInfo, 47
  - FlyCapture2Managed::GigEProperty, 67
- minNumImageNotifications
  - FlyCapture2Managed::FC2Config, 56

- minPacketSize
  - FlyCapture2Managed::Format7Info, 62
- minValue
  - FlyCapture2Managed::StrobeInfo, 148
- minor
  - FlyCapture2Managed::FC2Version, 59
- mode
  - FlyCapture2Managed::Format7-ImageSettings, 60
  - FlyCapture2Managed::Format7Info, 62
  - FlyCapture2Managed::TriggerMode, 163
- modeMask
  - FlyCapture2Managed::TriggerModeInfo, 164
- modelName
  - FlyCapture2Managed::CameraInfo, 42
- networkInterfaceIndex
  - FlyCapture2Managed::GigEStreamChannel, 69
- nodeNumber
  - FlyCapture2Managed::CameraInfo, 42
- nodeVendorId
  - FlyCapture2Managed::ConfigROM, 52
- numBanks
  - FlyCapture2Managed::LutData, 73
- numBuffers
  - FlyCapture2Managed::FC2Config, 56
- numChannels
  - FlyCapture2Managed::LutData, 73
- numCpuCores
  - FlyCapture2Managed::SystemInfo, 150
- numCurrents
  - FlyCapture2Managed::CameraStats, 50
- numEntries
  - FlyCapture2Managed::LutData, 73
- numImageNotifications
  - FlyCapture2Managed::FC2Config, 57
- numResendPacketsReceived
  - FlyCapture2Managed::CameraStats, 50
- numResendPacketsRequested
  - FlyCapture2Managed::CameraStats, 50
- numVoltages
  - FlyCapture2Managed::CameraStats, 50
- offsetHStepSize
  - FlyCapture2Managed::Format7Info, 62
  - FlyCapture2Managed::GigEImageSettingsInfo, 66
- offsetVStepSize
  - FlyCapture2Managed::Format7Info, 62
  - FlyCapture2Managed::GigEImageSettingsInfo, 66
- offsetX
  - FlyCapture2Managed::Format7-ImageSettings, 60
  - FlyCapture2Managed::GigEImageSettings, 65
- offsetY
  - FlyCapture2Managed::Format7-ImageSettings, 60
  - FlyCapture2Managed::GigEImageSettings, 65
- onOff
  - FlyCapture2Managed::CameraProperty, 44
  - FlyCapture2Managed::Embedded-ImageInfoProperty, 54
  - FlyCapture2Managed::StrobeControl, 147
  - FlyCapture2Managed::TriggerMode, 163
- onOffSupported
  - FlyCapture2Managed::CameraPropertyInfo, 47
  - FlyCapture2Managed::StrobeInfo, 148
  - FlyCapture2Managed::TriggerModeInfo, 164
- onePush
  - FlyCapture2Managed::CameraProperty, 44
- onePushSupported

- FlyCapture2Managed::Camera-PropertyInfo, [47](#)
- operator=
  - FlyCapture2Managed::ManagedPG-RGuid, [138](#)
- operator==
  - FlyCapture2Managed::ManagedPG-RGuid, [139](#)
- osDescription
  - FlyCapture2Managed::SystemInfo, [150](#)
- osType
  - FlyCapture2Managed::SystemInfo, [150](#)
- outputBitDepth
  - FlyCapture2Managed::LutData, [74](#)
- packetSize
  - FlyCapture2Managed::Format7Info, [62](#)
  - FlyCapture2Managed::GigEStream-Channel, [69](#)
- parameter
  - FlyCapture2Managed::TriggerMode, [163](#)
- pcieBusSpeed
  - FlyCapture2Managed::CameraInfo, [42](#)
- percentage
  - FlyCapture2Managed::Format7Info, [62](#)
- pixelFormat
  - FlyCapture2Managed::Format7-ImageSettings, [60](#)
  - FlyCapture2Managed::GigEImage-Settings, [65](#)
  - FlyCapture2Managed::Managed-Image, [135](#)
- pixelFormatBitField
  - FlyCapture2Managed::Format7Info, [63](#)
  - FlyCapture2Managed::GigEImage-SettingsInfo, [66](#)
- polarity
  - FlyCapture2Managed::Strobe-Control, [147](#)
  - FlyCapture2Managed::TriggerMode, [163](#)
- polaritySupported
- FlyCapture2Managed::StrobeInfo, [148](#)
- FlyCapture2Managed::TriggerMode-Info, [164](#)
- portErrors
  - FlyCapture2Managed::CameraStats, [50](#)
- present
  - FlyCapture2Managed::Camera-Property, [44](#)
  - FlyCapture2Managed::Camera-PropertyInfo, [47](#)
  - FlyCapture2Managed::StrobeInfo, [148](#)
  - FlyCapture2Managed::TriggerMode-Info, [164](#)
- progressive
  - FlyCapture2Managed::JpegOption, [71](#)
- propType
  - FlyCapture2Managed::GigEProperty, [67](#)
- ptr
  - FlyCapture2Managed::NativeEvent-Struct, [144](#)
- quality
  - FlyCapture2Managed::JpegOption, [72](#)
  - FlyCapture2Managed::Jpg2Option, [72](#)
- readOutSupported
  - FlyCapture2Managed::Camera-PropertyInfo, [47](#)
  - FlyCapture2Managed::StrobeInfo, [149](#)
  - FlyCapture2Managed::TriggerMode-Info, [165](#)
- receivedDataSize
  - FlyCapture2Managed::Managed-Image, [135](#)
- recommendedBytesPerPacket
  - FlyCapture2Managed::Format7-PacketInfo, [63](#)
- regReadFailed
  - FlyCapture2Managed::CameraStats, [50](#)
- regWriteFailed

- FlyCapture2Managed::CameraStats, 50
- registerTimeout
  - FlyCapture2Managed::FC2Config, 57
- registerTimeoutRetries
  - FlyCapture2Managed::FC2Config, 57
- rows
  - FlyCapture2Managed::Managed-Image, 135
- screenHeight
  - FlyCapture2Managed::SystemInfo, 150
- screenWidth
  - FlyCapture2Managed::SystemInfo, 150
- seconds
  - FlyCapture2Managed::TimeStamp, 153
- sensorInfo
  - FlyCapture2Managed::CameraInfo, 42
- sensorResolution
  - FlyCapture2Managed::CameraInfo, 42
- serialNumber
  - FlyCapture2Managed::CameraInfo, 42
- shutter
  - FlyCapture2Managed::Embedded-ImageInfo, 54
- softwareTriggerSupported
  - FlyCapture2Managed::TriggerMode-Info, 165
- source
  - FlyCapture2Managed::Strobe-Control, 147
  - FlyCapture2Managed::StrobeInfo, 149
  - FlyCapture2Managed::TriggerMode, 163
- sourceMask
  - FlyCapture2Managed::TriggerMode-Info, 165
- sourcePort
  - FlyCapture2Managed::GigEStream-Channel, 69
- stride
  - FlyCapture2Managed::Managed-Image, 135
- strobePattern
  - FlyCapture2Managed::Embedded-ImageInfo, 54
- subnetMask
  - FlyCapture2Managed::CameraInfo, 42
- supported
  - FlyCapture2Managed::LutData, 74
- systemInfo
  - FlyCapture2Managed::Managed-Utilities, 144
- systemMemorySize
  - FlyCapture2Managed::SystemInfo, 151
- temperature
  - FlyCapture2Managed::CameraStats, 50
- timeSinceBusReset
  - FlyCapture2Managed::CameraStats, 51
- timeSinceInitialization
  - FlyCapture2Managed::CameraStats, 51
- timeStamp
  - FlyCapture2Managed::CameraStats, 51
  - FlyCapture2Managed::Managed-Image, 135
- timestamp
  - FlyCapture2Managed::Embedded-ImageInfo, 54
- translate
  - FlyCapture2Managed::Translate, 160–162
- type
  - FlyCapture2Managed::Camera-Property, 45
  - FlyCapture2Managed::Camera-PropertyInfo, 47
  - FlyCapture2Managed::FC2Version, 59
- unitAbbr
  - FlyCapture2Managed::Camera-PropertyInfo, 47
- unitBytesPerPacket

- FlyCapture2Managed::Format7-  
PacketInfo, [63](#)
- unitSWVer
  - FlyCapture2Managed::ConfigROM,  
[52](#)
- unitSpecId
  - FlyCapture2Managed::ConfigROM,  
[52](#)
- unitSubSWVer
  - FlyCapture2Managed::ConfigROM,  
[52](#)
- units
  - FlyCapture2Managed::Camera-  
PropertyInfo, [47](#)
- userDefinedName
  - FlyCapture2Managed::CameraInfo,  
[43](#)
- value
  - FlyCapture2Managed::GigEProperty,  
[68](#)
- value0
  - FlyCapture2Managed::ManagedPG-  
RGuid, [139](#)
- value1
  - FlyCapture2Managed::ManagedPG-  
RGuid, [139](#)
- value2
  - FlyCapture2Managed::ManagedPG-  
RGuid, [139](#)
- value3
  - FlyCapture2Managed::ManagedPG-  
RGuid, [139](#)
- valueA
  - FlyCapture2Managed::Camera-  
Property, [45](#)
- valueB
  - FlyCapture2Managed::Camera-  
Property, [45](#)
- valueReadable
  - FlyCapture2Managed::TriggerMode-  
Info, [165](#)
- vendorName
  - FlyCapture2Managed::CameraInfo,  
[43](#)
- vendorPixelFormatBitField
  - FlyCapture2Managed::Format7Info,  
[63](#)
  - FlyCapture2Managed::GigEImage-  
SettingsInfo, [66](#)
- vendorUniqueInfo0
  - FlyCapture2Managed::ConfigROM,  
[52](#)
- vendorUniqueInfo1
  - FlyCapture2Managed::ConfigROM,  
[52](#)
- vendorUniqueInfo2
  - FlyCapture2Managed::ConfigROM,  
[52](#)
- vendorUniqueInfo3
  - FlyCapture2Managed::ConfigROM,  
[52](#)
- whiteBalance
  - FlyCapture2Managed::Embedded-  
ImageInfo, [54](#)
- width
  - FlyCapture2Managed::Format7-  
ImageSettings, [60](#)
  - FlyCapture2Managed::GigEImage-  
Settings, [65](#)
- xmlURL1
  - FlyCapture2Managed::CameraInfo,  
[43](#)
- xmlURL2
  - FlyCapture2Managed::CameraInfo,  
[43](#)